

Data and Modeling

What makes something a statistical model?

a math model that takes into randomness in the data

- ex: Linear Regression

random error

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

response explanatory variable (predictor)

What is the difference between prediction and inference?

modeling a 'new' value

- either predicting a new response

or classifying a new observation

Statements or claims about relationships in the population

using sample data

Data

- When modeling, what should our data look like?

2-D table

Variables

observations

	A	B	C	D
1				
2				
3				
4				
:				
n				

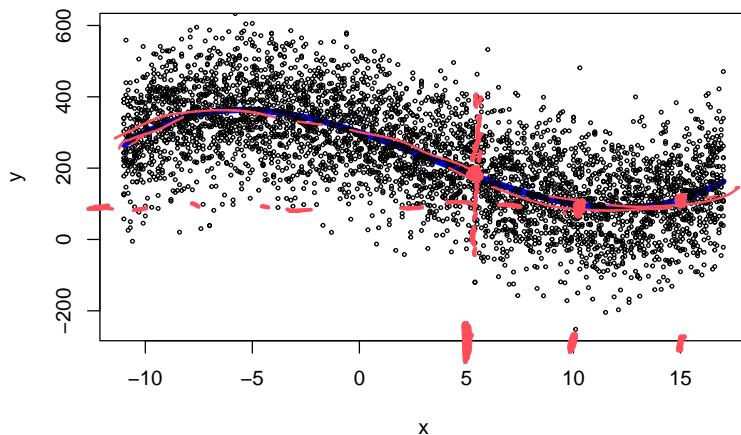
Relating Explanatory Variables to a Response Variable

Consider the response Y as a random variable. We'll consider the x values fixed (for any explanatory variable). Our interest is in learning about the relationship between Y and x .

Y is random, so we don't have a **deterministic** relationship...

*T
random
- observed
differently
each time
we see it*

Below: Blue line, $f(x)$, is the 'true' relationship between x and y

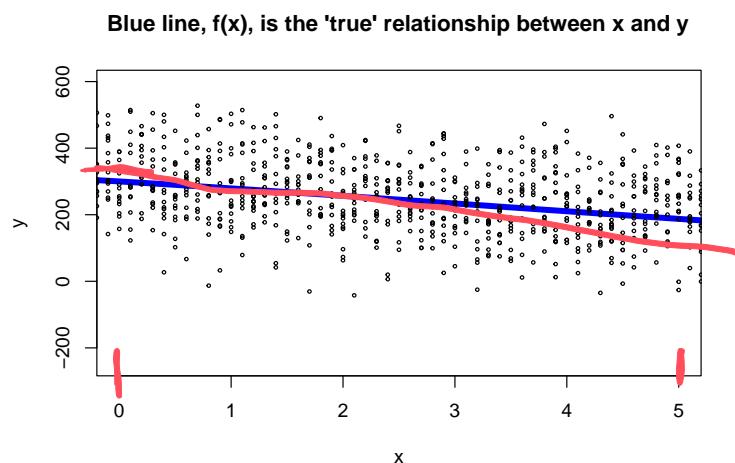


What should we try to relate/model?

$$f(x) = E(Y|x) \quad \text{given}$$
$$E(Y|x=10) = 100$$

Approximating $f(x)$

Although the true relationship is most certainly nonlinear, we may be ok approximating the relationship linearly. For example, consider the same plot as above but between 0 and 5 only:



That's pretty linear. Consider plot between 5 and 15:

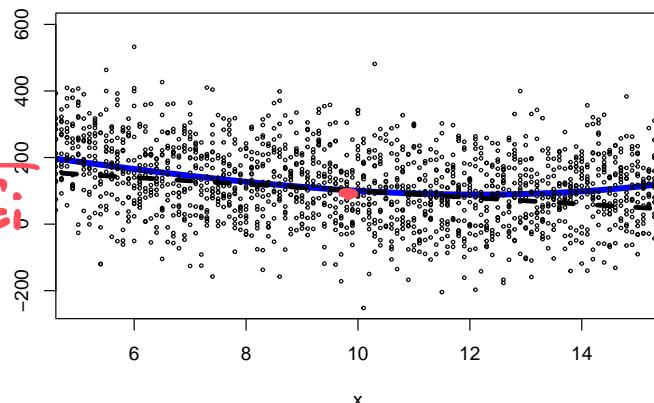
Quadratic Linear Regression

• Blue line, $f(x)$, is the 'true' relationship between x and y
Dashed line is the linear approximation

L_M the predictor

$$Y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$$

$$Y_i = \beta_0 e^{-\beta_1 x_i}$$



Line still does a reasonable job and is often used as a basic approximation.

Exploratory Data Analysis (EDA)

What are our first steps with data?

get to know your data, make sure it is what you think
it should be, inform model choices

Common steps to EDA

1. Read in your data
2. Understand how your data is stored
3. Understand missing data values (NA)
4. Clean up data
5. Summarize the data (min, max, mean, frequency/count, scatter plot, bar graph, histogram)
6. Transformations? Repeat S

Data Intro

This dataset contains information about used motorcycles and their cost.

From the information page: This data can be used for a lot of purposes such as price prediction to exemplify the use of linear regression in Machine Learning. The columns in the given dataset are as follows:

- name
- selling price *response*
- year
- seller type
- owner
- km driven
- ex showroom price

The data are available to download from this URL:

<https://www4.stat.ncsu.edu/~online/datasets/bikeDetails.csv>

Read in Data and Explore!

```
library(tidyverse)
bikeData <- read_csv("https://www4.stat.ncsu.edu/~online/datasets/bikeDetails.csv")
select(bikeData, selling_price, year, km_driven, ex_showroom_price, name, everything())

## # A tibble: 1,061 x 7
##   selling_price    year  km_driven ex_showroom_price name      seller_type owner
##       <dbl>     <dbl>      <dbl>          <dbl> <chr>      <chr>    <chr>
## 1        175000 2019         350            NA Royal Enfi~ Individual 1st ~
## 2         45000 2017         5650           NA Honda Dio  Individual 1st ~
## 3        150000 2018        12000          148114 Royal Enfi~ Individual 1st ~
## 4         65000 2015        23000          89643 Yamaha Faz~ Individual 1st ~
## 5         20000 2011        21000           NA Yamaha SZ ~ Individual 2nd ~
## 6         18000 2010        60000          53857 Honda CB T~ Individual 1st ~
## 7         78500 2018        17000          87719 Honda CB H~ Individual 1st ~
## 8        180000 2008        39000           NA Royal Enfi~ Individual 2nd ~
## 9         30000 2010        32000           NA Hero Honda~ Individual 1st ~
## 10        50000 2016        42000          60122 Bajaj Disc~ Individual 1st ~
## # ... with 1,051 more rows
```

Our ‘response’ variable here is the `selling_price` and we could use the variable `year`, `km_driven`, or `ex_showroom_price` as the explanatory variable. Let’s make some plots and summaries to explore.

Linear Regression

Recap: Our goal is to predict a value of Y while including an explanatory variable x . We are assuming we have a sample of (x_i, y_i) pairs, $i = 1, \dots, n$.

The Simple Linear Regression (SLR) model can be used:

$$Y_i = \beta_0 + \beta_1 x_i + E_i$$

where

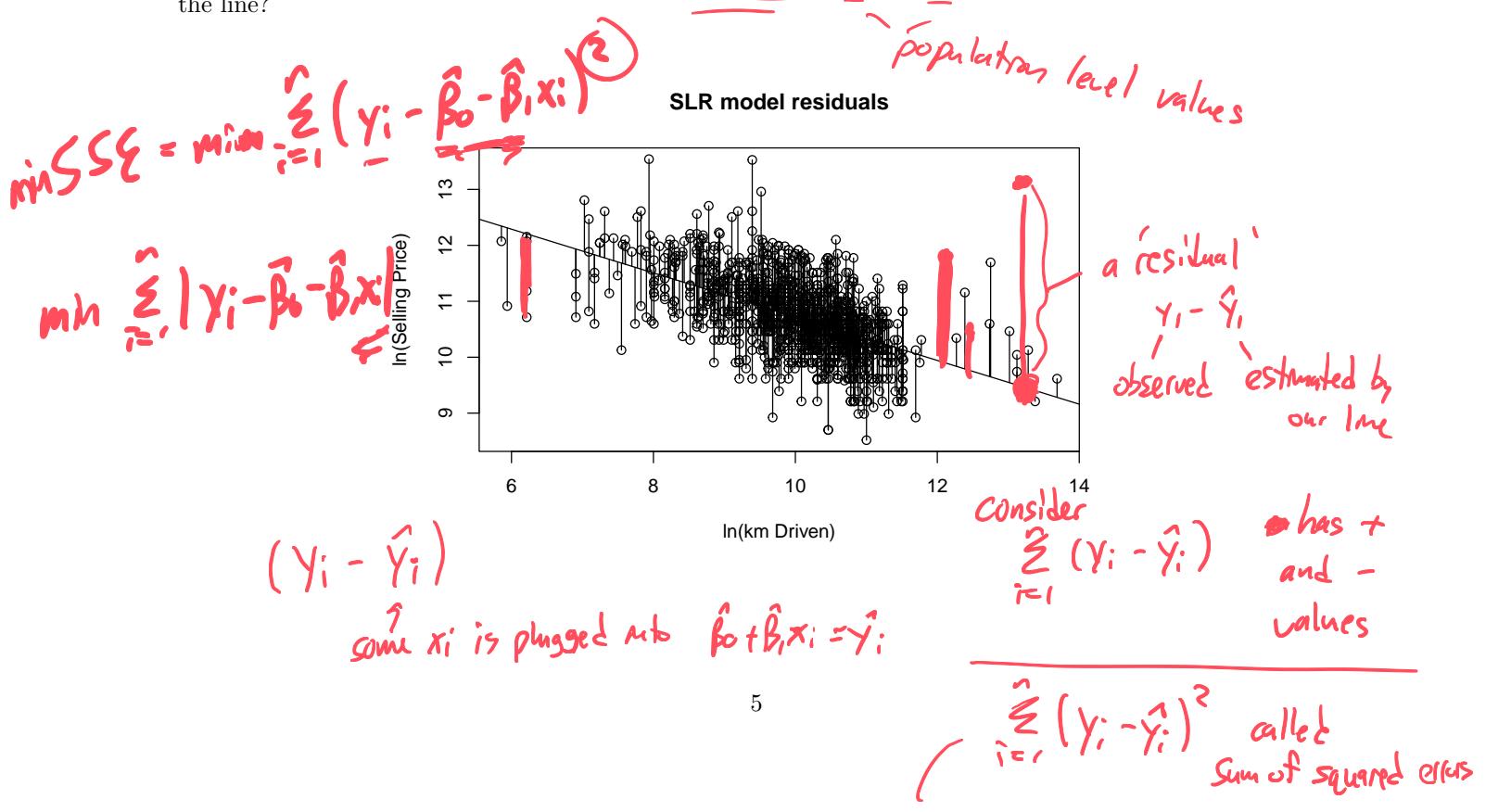
- Y_i is our response for the i^{th} observation
- x_i is the value of our explanatory variable for the i^{th} observation
- β_0 is the y intercept
- β_1 is the slope
- $E_i \stackrel{iid}{\sim} N(0, \sigma^2)$

Assume E_1 is independent of E_2 $P(A|B) = P(A)$

What is important to know from all that??

- consider 'distance' between 'fitted' curve and data points
- estimate the SD from that
- modeling relationship with a line
- Slope very important, $\beta_1 = 0$ our 'test' of interest?

We fit this model to data. That is, find the best estimators of β_0 and β_1 (and σ^2) given the data. How to fit the line?



$$L = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

minimize this with respect to
 $\hat{\beta}_0, \hat{\beta}_1$

Fitting the line

The (fitted) linear regression model uses $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$. Calculus allows us to find the 'least squares' estimators, $\hat{\beta}_0$ and $\hat{\beta}_1$ in a nice closed-form!

$$\begin{aligned} \frac{d}{d\hat{\beta}_0} \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 &= \sum_{i=1}^n 2(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)'(-1) \quad \text{set equal to } 0 \\ \frac{d}{d\hat{\beta}_1} \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 &= \sum_{i=1}^n 2(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)'(-x_i) \quad \text{set equal to } 0 \\ \hat{\beta}_1 &= \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} \end{aligned}$$

$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$
 $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$

Making Inference

What hypothesis are we interested in and why?

$$H_0: \beta_1 = 0$$

$\beta_1 = 0$ implies we have a constant relationship between x and y

$$H_A: \beta_1 \neq 0$$

$$Y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \epsilon_i$$

H0: $\beta_1 = \beta_2 = 0$
 H1: At least 1 is not 0

$$\underline{Y_i = \beta_0 + \epsilon_i}$$

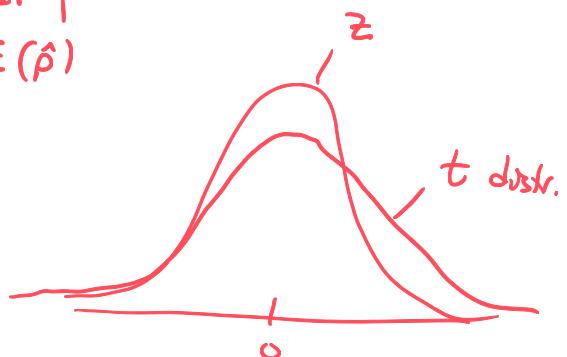
How can we form a confidence interval for the quantity of interest?

Recall: $\hat{p} \pm z^* \sqrt{\frac{p(1-p)}{n}}$ is a CI for p

$$\hat{\beta}_1 \pm z^* SE(\hat{\beta}_1)$$

almost ... instead

$$\hat{\beta}_1 \pm t^* S\hat{E}(\hat{\beta}_1)$$



Checking assumptions

How can we check our assumptions on the errors?

- Consider the observed residuals $y_i - \hat{y}_i \quad i=1, \dots, n$
- plot a histogram of these!
 - density plot (smoothed version)
 - qq plot
 - look for straight line

Fitting a Linear Regression Model in R

We can fit the model with the `lm()` function. Provide a formula

$response \sim explanatory\ variable\ equation$ (intercept fit by default)

```
library(tidyverse)
bikeData <- read_csv("https://www4.stat.ncsu.edu/~online/datasets/bikeDetails.csv")
bikeData <- bikeData %>% mutate(log_selling_price = log(selling_price),
                                log_km_driven = log(km_driven))

fit <- lm(log_selling_price ~ log_km_driven, data = bikeData) can add more explanatory variables here
```

R object *Y* *X*

Determine the fitted model by looking at the `coefficients` element.

```
fit$coefficients
```

β_0 β_1

(Intercept) log_km_driven
14.6355683 -0.3910865

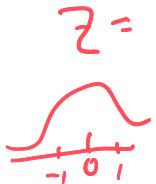
far enough from 0? look at test or CI

Look at the hypothesis test of interest with `summary()`

```
summary(fit)
```

```
##  
## Call:  
## lm(formula = log_selling_price ~ log_km_driven, data = bikeData)  
##
```

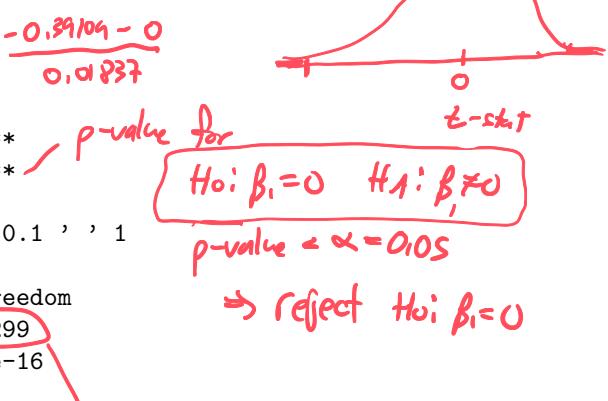
$$Z = \frac{\text{stat} - \text{mean (true mean)}}{\text{SD (stat)}}$$



$p\text{-value} = P(\text{get a sample slope as far or further away from } 0, \text{ if } \beta_1 \text{ was truly } 0)$

```
## Residuals:
##      Min     1Q Median     3Q    Max
## -1.9271 -0.3822 -0.0337  0.3794  2.5656
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.63557   0.18455 79.31 <2e-16 ***
## log_km_driven -0.39109  0.01837 -21.29 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5953 on 1059 degrees of freedom
## Multiple R-squared:  0.2997, Adjusted R-squared:  0.299
## F-statistic: 453.2 on 1 and 1059 DF, p-value: < 2.2e-16
```

What here is important and why?



closer to 1
is better

Find a confidence interval with `confint()`

```
confint(fit)
```

```
##              2.5 %    97.5 %
## (Intercept) 14.2734501 14.9976864
## log_km_driven -0.4271342 -0.3550389
```

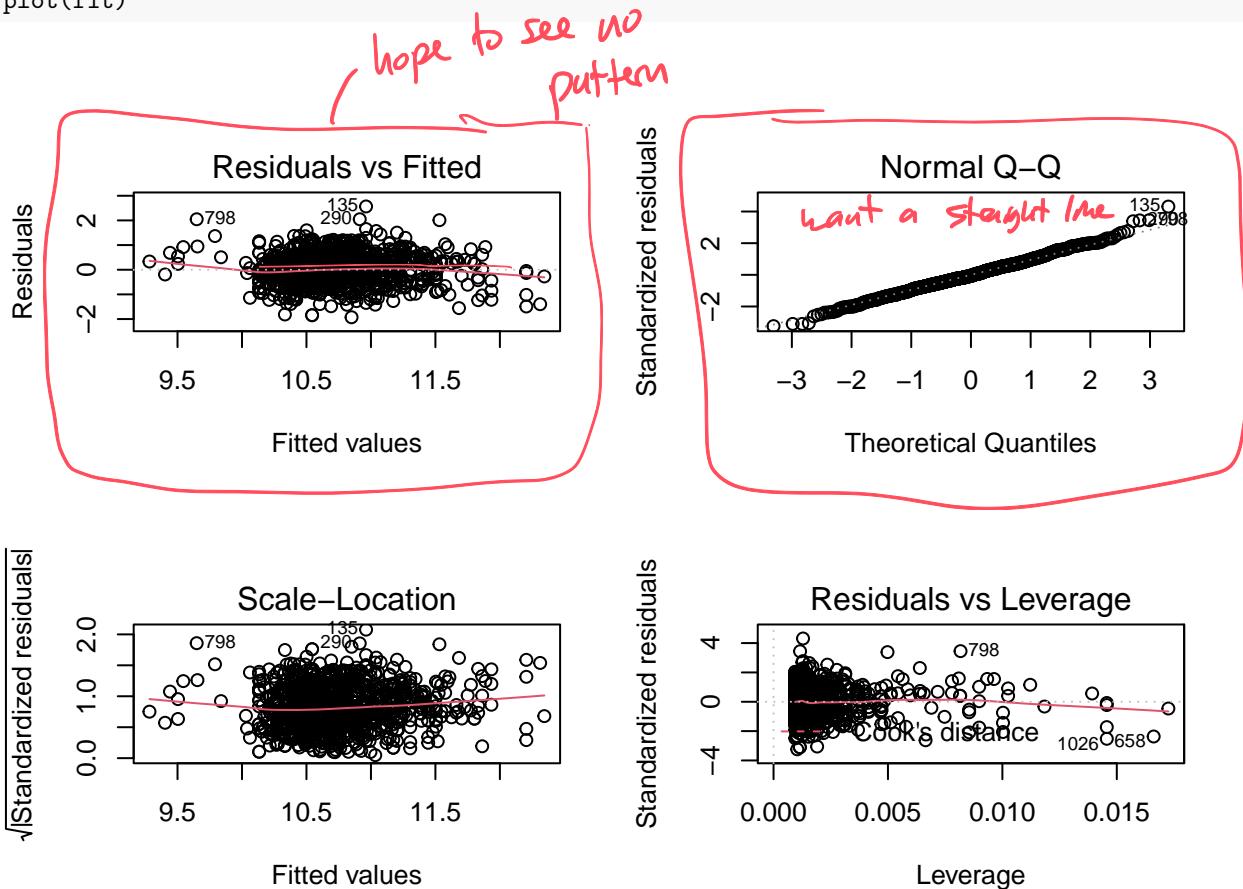
95% CI for β_1

0 is not in here \Rightarrow 0 not plausible \Rightarrow significant relationship

between (log)
selling price and
km driven

Check conditions! `plot()` on the model fit will work.

```
par(mfrow = c(2,2))
plot(fit)
```



Logistic Regression Model

response ~ Var1 + Var2

Used when you have a **binary** response variable

- Using SLR is not appropriate!

Example:

- Consider data about **water potability**

```
library(tidyverse)
water <- read_csv("water_potability.csv")
water
```

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon		
## 1	NA	205.	20791.	7.30	369.	564.	10.4		
## 2	3.72	129.	18630.	6.64	NA	593.	15.2		
## 3	8.10	224.	19910.	9.28	NA	419.	16.9		
## 4	8.32	214.	22018.	8.06	357.	363.	18.4		
## 5	9.09	181.	17979.	6.55	310.	398.	11.6		
## 6	5.58	188.	28749.	7.54	327.	280.	8.40		
## 7	10.2	248.	28750.	7.51	394.	284.	13.8		
## 8	8.64	203.	13672.	4.56	303.	475.	12.4		
## 9	NA	119.	14286.	7.80	269.	389.	12.7		
## 10	11.2	227.	25485.	9.08	404.	564.	17.9		
## # ... with 3,266 more rows, and 3 more variables: Trihalomethanes <dbl>, Turbidity <dbl>, Potability <dbl>									

- Summarize water potability

water\$potability <- as.factor(water\$Potability)

```
table(water$Potability)
```

levels of ↑
 ##
 ## 0 1
 ## 1998 1278

not *drinking*

```
water %>%
```

```
group_by(Potability) %>%  

  select(Hardness, Chloramines) %>%
```

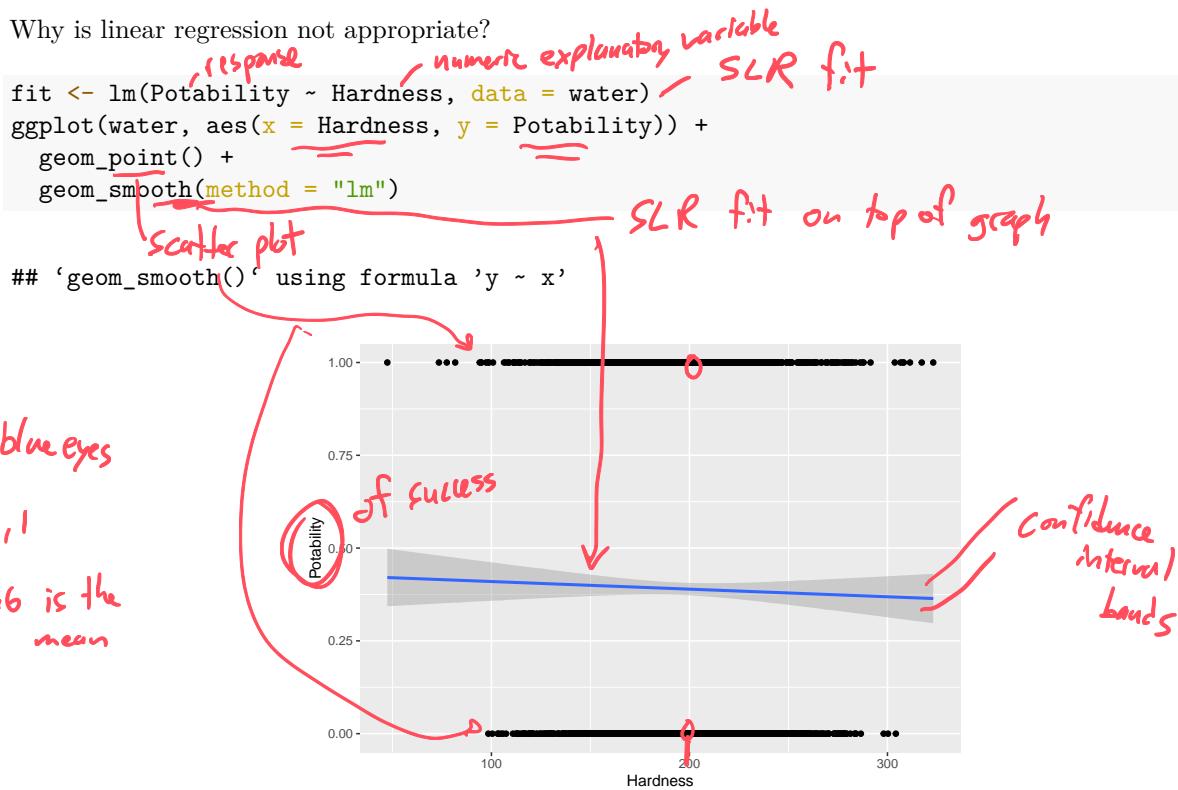
```
summarise()
```

Summarize (meanH = mean(Hardness), meanC = mean(Chloramines))

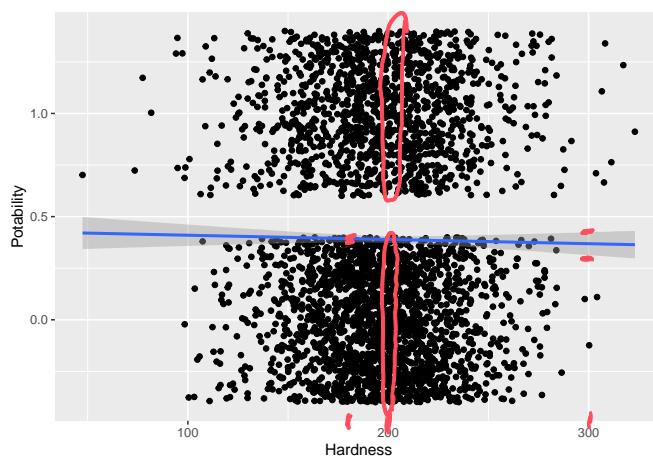
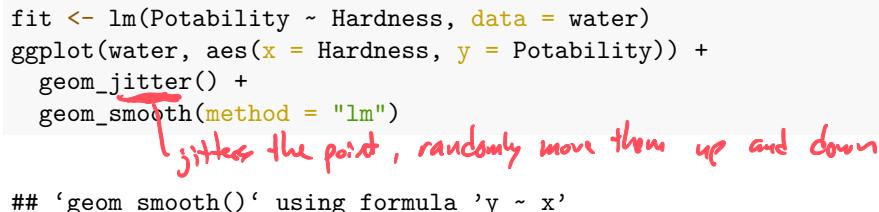
```
## Adding missing grouping variables: 'Potability'
```

Potability	Hardness	Chloramines
Min.	Min. : 47.43	Min. : 0.352
1st Qu.	1st Qu.:176.85	1st Qu.: 6.127
Median	Median :196.97	Median : 7.130
Mean	Mean :196.37	Mean : 7.122
3rd Qu.	3rd Qu.:216.67	3rd Qu.: 8.115
Max.	Max. :323.12	Max. :13.127

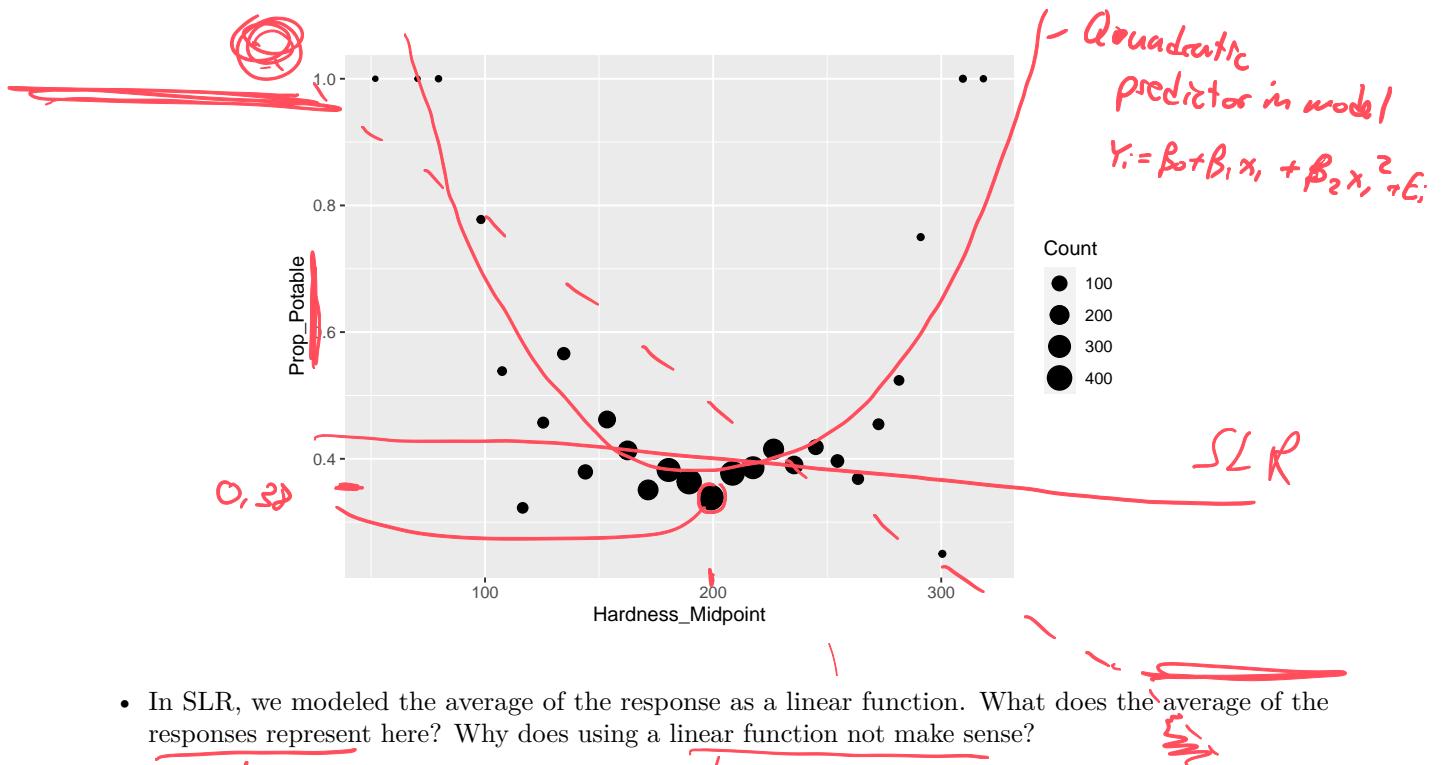
Why is linear regression not appropriate?



Better view...



A better view of the data is to visualize the proportions of successes as a function of hardness.



- In SLR, we modeled the average of the response as a linear function. What does the average of the responses represent here? Why does using a linear function not make sense?

Estimated probability of Success (probability)

→ A line will eventually predict a probability outside of $[0, 1]$

- Basic Logistic Regression models success probability using the *logistic function*

$$P(\text{success}|x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

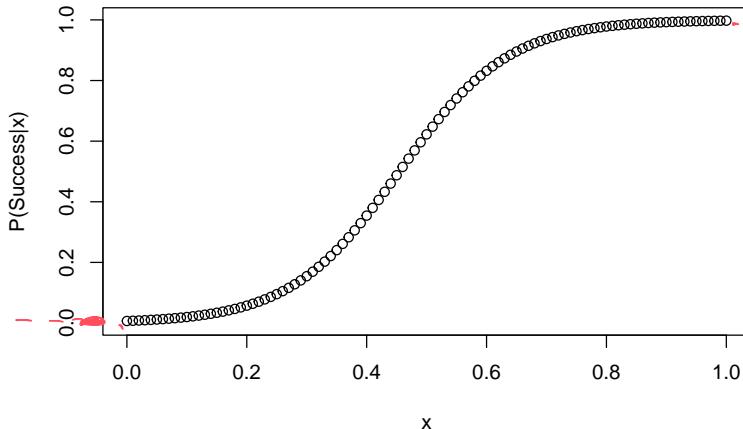
*E(Y|x) - population mean
given*

*non linear
function of our predictor*

- This function never goes below 0 and never above 1 - works great for many applications!
- The logistic regression model doesn't have a closed form solution (maximum likelihood often used to fit parameters)

$$\hat{\beta}_1 = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sum (x_i - \bar{x})^2}$$

$$b_0 = -5, b_1 = 11$$



- Back-solving shows the *logit* or *log-odds* of success is linear in the parameters

$$\log \left(\frac{P(\text{success}|x)}{1 - P(\text{success}|x)} \right) = \beta_0 + \beta_1 x$$

P(Failure|x)

- Coefficient interpretation changes greatly from linear regression model!
- β_1 represents a change in the log-odds of success

for a one unit change in x

Hypotheses of Interest

What do you think would indicate that x is related to the probability of success here?

$$H_0: \beta_1 = 0 \quad \text{vs} \quad H_A: \beta_1 \neq 0$$

$$P(\text{Success}|x) = \frac{e^{\beta_0}}{1 + e^{\beta_0}} \quad \text{if } \beta_1 = 0$$

Fitting a Logistic Regression Model in R

Fit in R using `glm()` with `family = binomial` and a formula just like `lm()`.

```
fit <- glm(Potability ~ Hardness, data = water, family = "binomial")
```

Get coefficients by looking at `coefficients` element:

```
fit$coefficients
```

```
## (Intercept) Hardness
## -0.2774792831 -0.0008629619
```

Get hypothesis test via `summary()`:

```
summary(fit)
```

```
##
## Call:
## glm(formula = Potability ~ Hardness, family = "binomial", data = water)
##
## Deviance Residuals:
##    Min      1Q      Median      3Q      Max
## -1.0279 -0.9963 -0.9853  1.3678  1.4209
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.277479   0.216758 -1.280   0.200
## Hardness     -0.000863   0.001090 -0.792   0.428
## 
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 4382.0 on 3275 degrees of freedom
## Residual deviance: 4381.3 on 3274 degrees of freedom
## AIC: 4385.3
##
## Number of Fisher Scoring iterations: 4
```

Get confidence interval for β_1 with:

```
confint(fit)
```

```
## Waiting for profiling to be done...
##
##              2.5 %      97.5 %
## (Intercept) -0.702803063 0.147169863
## Hardness     -0.003000628 0.001272738
```

If we want a probability estimate back, use `predict()` with `type = 'link'`:

~~response~~

$$H_0: \beta_1 = 0$$

$$H_A: \beta_1 \neq 0$$

Fail to reject H_0

\circ is in here

values of x you want prediction for

```

predict(fit, newdata = data.frame(Hardness = c(200, 300)), type = "link", se.fit = TRUE)

## $fit
##      1      2
## -0.4500717 -0.5363679
##
## $se.fit
##      1      2
## 0.03606504 0.11869267
##
## $residual.scale
## [1] 1

```

Visualize the fit:

