Data Cleaning, Bayes

Lecture 14

Connor Dowd

May 13th, 2021

Today's Class

1. Review
   - Cleaning Rules of Thumb
2. Data Cleaning – a bit more
   - Outliers
   - Winsorizing
   - Censored Data
3. Bayes
4. Predictions 2 - pt 2
   - CI vs PI
   - Model Uncertainty
5. HW5

Review

Data Cleaning 101

Rules of Thumb:

1. Keep looking at the data
2. Make small changes.
3. Test your changes before you overwrite variables.
4. Don't overwrite actual files unless you're certain.
5. **Don't throw away potentially useful data.**

Throwing Away Data:

When are we tempted to throw out observations?

- Duplicates

Throwing Away Data:

When are we tempted to throw out observations?

- Duplicates
- Missing Values

Throwing Away Data:

When are we tempted to throw out observations?

- Duplicates
- Missing Values
- Outliers?

Merge Problems: Duplicates

Throwing away duplicates could cause problems. What kind of problems?

If homes that sell frequently are fundamentally different from those that don't, and are important to our target questions, then throwing them out may bias our whole procedure.

What then? We need to know if duplicates reflect real-world differences. Did someone accidentally copy a row of data, or are duplicates a thing that happens in the world?

- IF they happen in the world – don't throw them away.

Dropping NAs

Dropping `NA` observations tends to be justified by the following assumption:

- The data is missing *at random*

This is frequently implausible.

Instead, what you should do is also model the `NA`s.

```
table(prop2na$fireplacecnt,useNA="ifany")
```

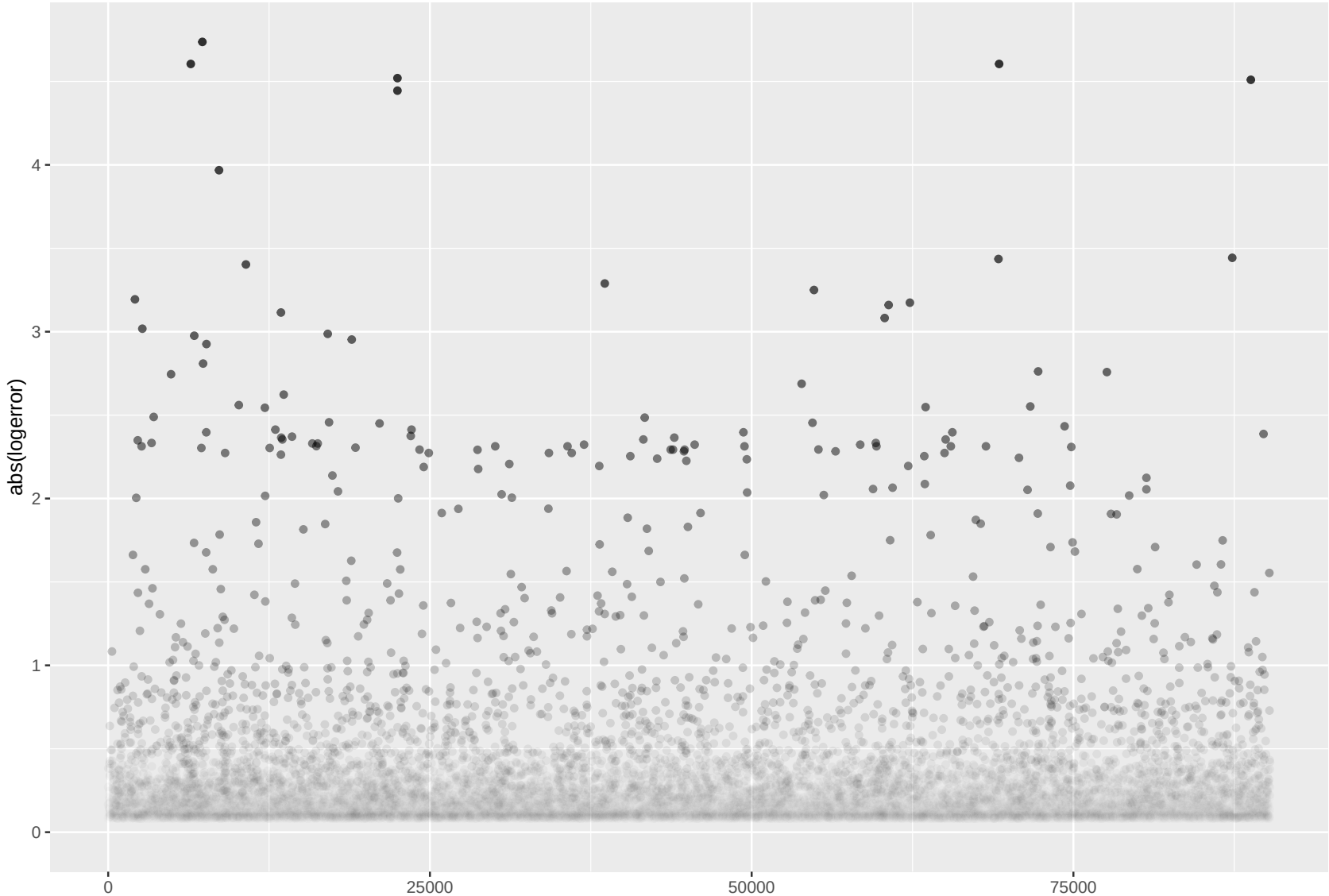
```
##
##    -1     1     2     3     4     5
## 80668  8165  1106   312    21     3
```

Outliers

What is an Outlier?

- Basic idea: Some observations are extreme, perhaps uninformative, perhaps should be removed.
  - This is mostly madness. But you need *some* familiarity with common forms of madness.

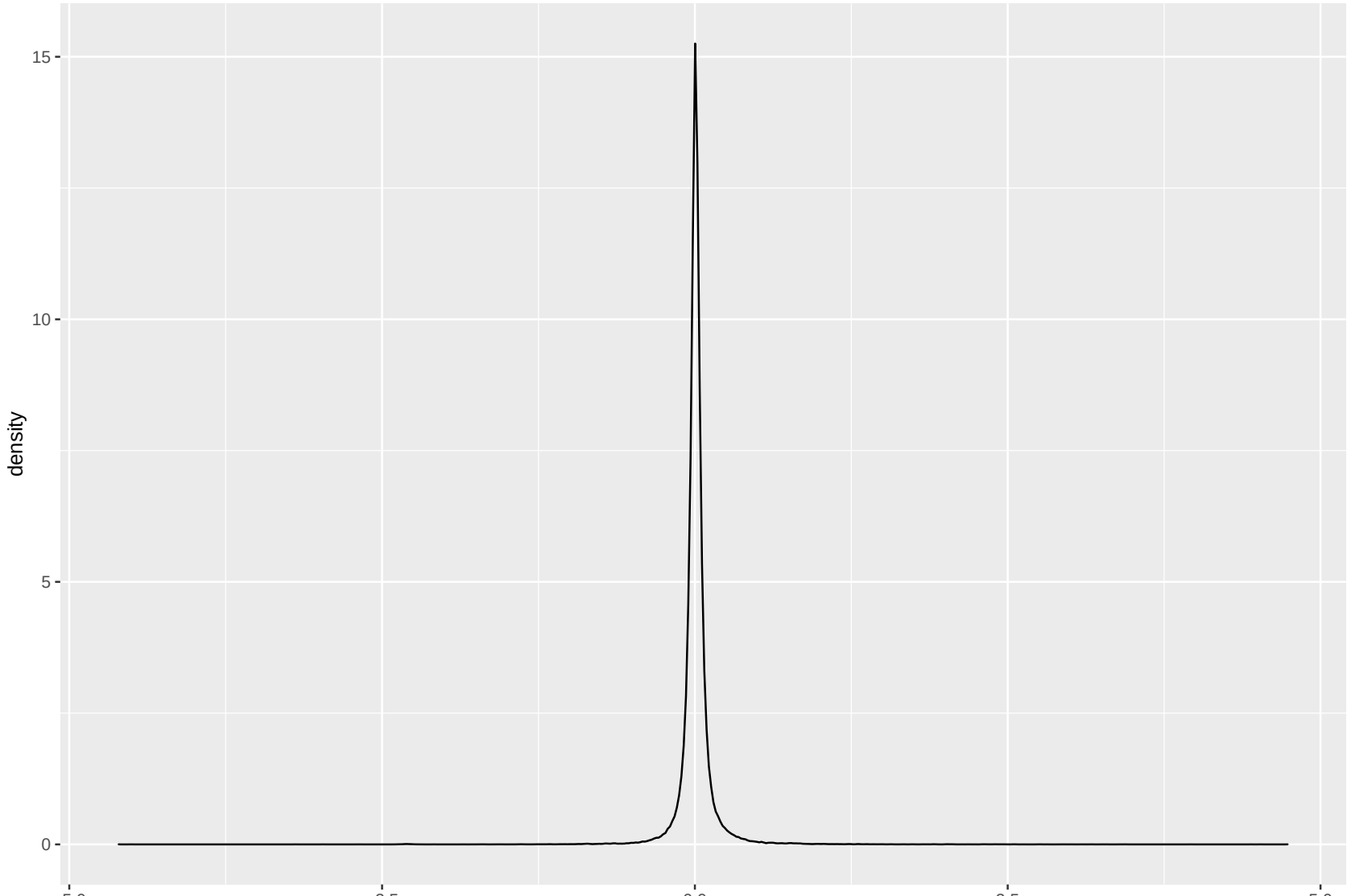
Outliers



- 6 of 90k observations are bigger than 4.
- 1/15000.

Outliers

```
ggplot(prop2na,aes(x=logerror))+
  geom_density()
```



Outliers

There are a few possibilities:

- Measurement error
- Clerical Error
- Genuine extreme observation

If you have *good reason* ahead of time, to think some observations are clerical errors or mismeasured, getting rid of them is a good idea.

If you are merely looking at observation's extremity and concluding they are clerical errors – you are playing a dangerous game.

Outliers

The question becomes the following.

Is it unreasonable that a 1-in-15000 observation would be above 4?

In general we don't know. For home prices, the answer is probably "no".

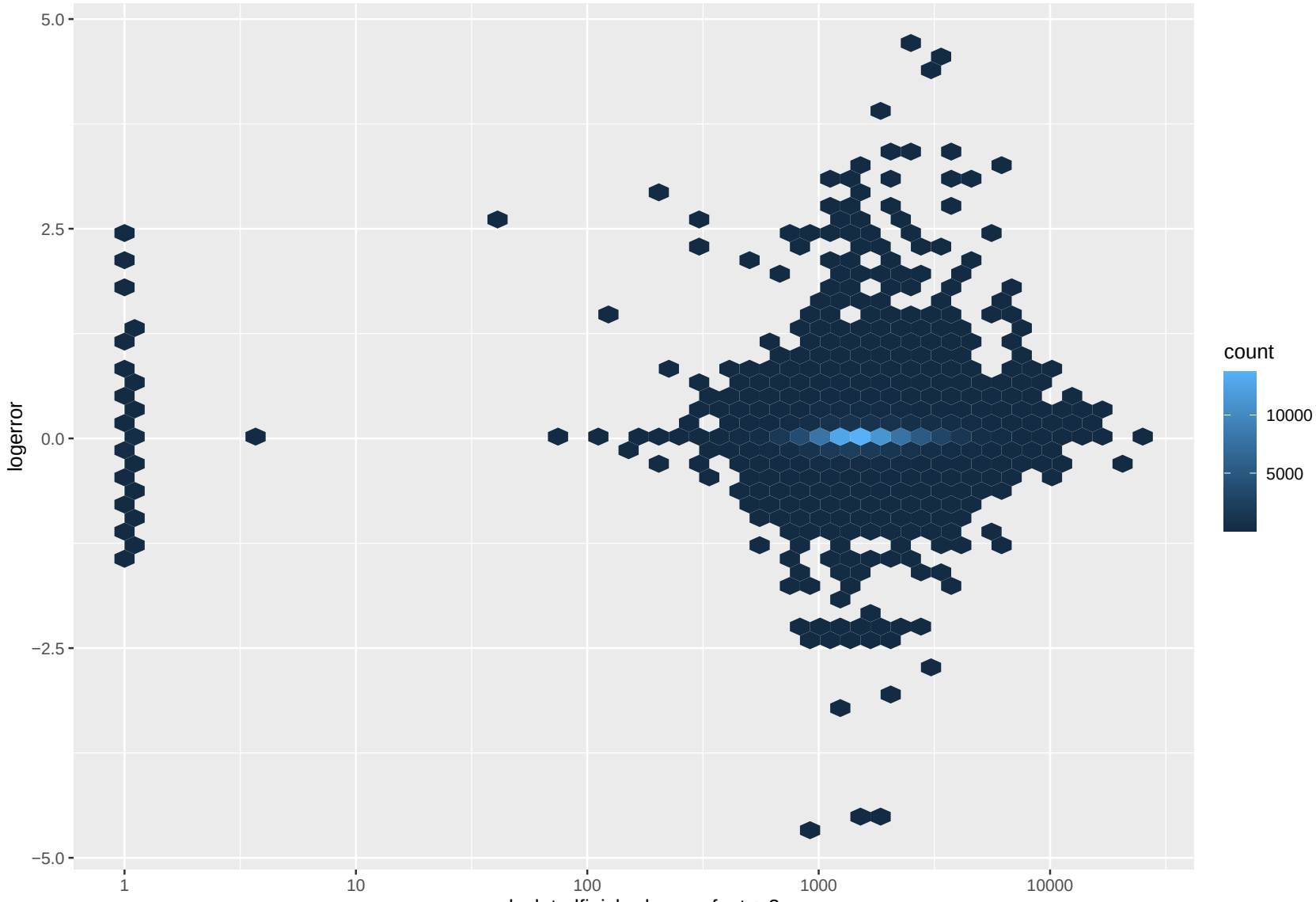
In some domains we do know a lot. E.g. if you're working with data on people's heights, we can rule out 20 feet. We can rule out 10 feet.

But in many domains there are *fat tails*.

Outliers

Why are outliers a problem at all?

- Suppose the observations are fake.



Outliers Influence

Outliers have a big influence on our coefficients.

This is because we often use loss functions which increase faster than the scale of the loss. E.g. Squared Error:

$$l_2(\hat{y}, y) = (\hat{y} - y)^2$$

Thus, when the size of an error doubles, our loss quadruples.

Or when the size of an error increases by 1, our loss increases by an amount equal to the already existing amount of error.

Outliers - Problems

The influence outliers have on our coeffiencts translates into influence on predictions.

Thus, non-real outliers can have a severe damaging effect on our predictions.

Importantly too – even in largeish samples, real outliers can cause our models to overfit.

Solutions?

The standard solution is to remove outliers. But what if the outliers were real?

There really were massive prediction errors for some home values. There really are people with *extremely* high incomes. Etc.

In domains where there are fat-tails – that is to say where some values are orders of magnitude larger than others – the extreme values don't just have an extreme influence on our predictions and inference, *they also have extreme influence on the true underlying means*.

Getting rid of them may well do more harm than good. Remember, they may be a real feature of the world.

Outliers

So what do we do?

We are worried that they have an extreme influence on our regressions. We are worried they may cause overfit. But we don't want to forget about them or ignore them completely.

One solution: *Winsorizing*

Winsorizing

Winsorizing is a simple idea. We don't want to drop extreme values. But we also don't like how extreme they are.

- What if we just made them less extreme?

Winsorizing

The standard routine does the following.

1. Pick some quantile (e.g. 1%).
2. Find that quantile – e.g. find the 99%ile abs(logerror)
3. Set values higher than that quantile to that quantile.

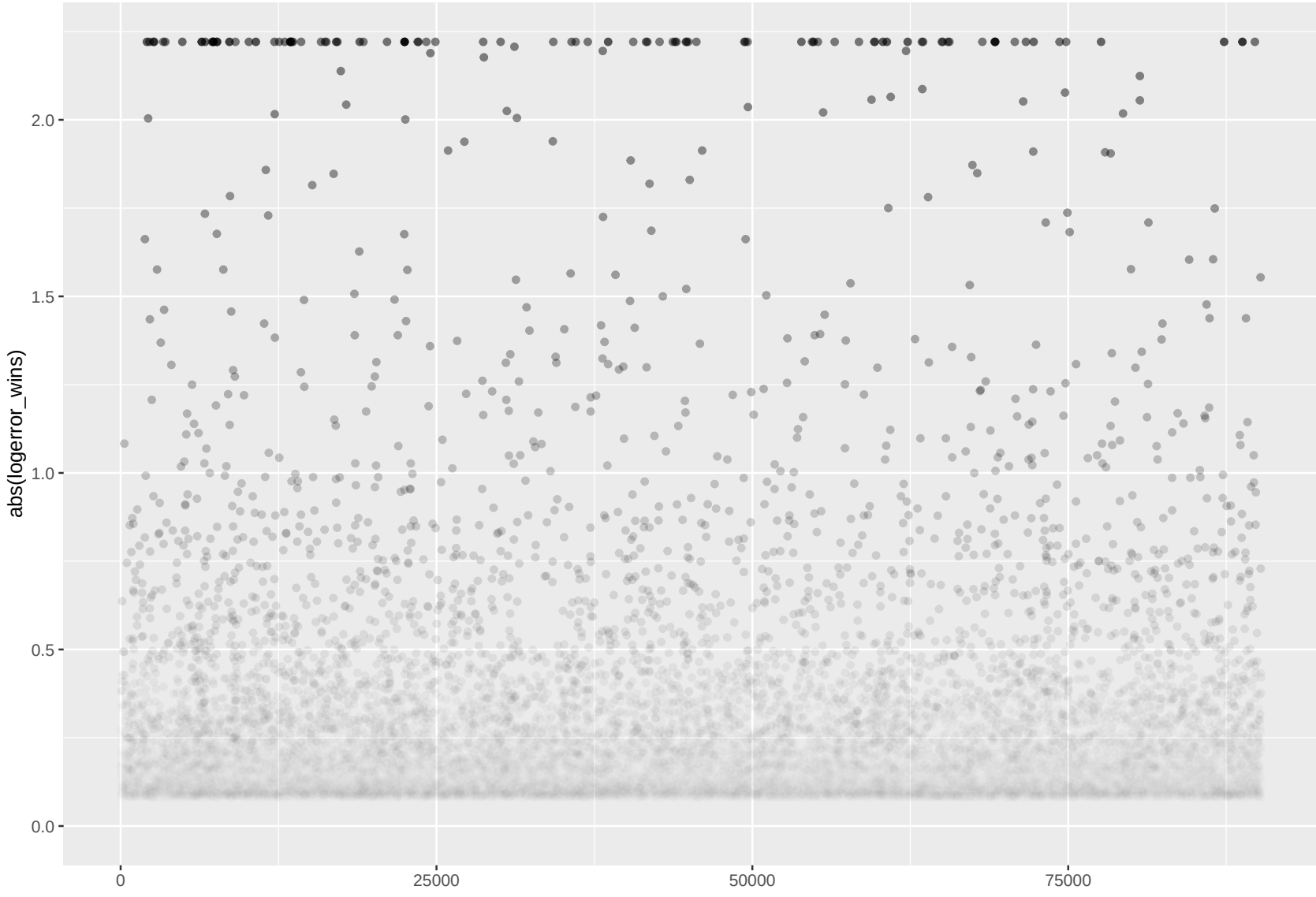
Winsorizing

```
quant = 0.001 #1-in-1000 or 0.1%
threshold = quantile(abs(prop2na$logerror),1-quant)
threshold
```

```
##     99.9%
## 2.220794
```

```
prop2na = prop2na %>%
  mutate(logerror_wins = ifelse(abs(logerror) > threshold,
                                sign(logerror)*threshold,
                                logerror))
```

Winsorized



Winsorizing

```
mean(abs(prop2na$logerror))
```

```
## [1] 0.06844671
```

```
mean(abs(prop2na$logerror_wins))
```

```
## [1] 0.06799503
```

```
var(prop2na$logerror)
```

```
## [1] 0.02594639
```

```
var(prop2na$logerror_wins)
```

```
## [1] 0.02335868
```

Winsorizing

Some statistics are unaffected.

```
median(abs(prop2na$logerror))
```

```
## [1] 0.0325
```

```
median(abs(prop2na$logerror_wins))
```

```
## [1] 0.0325
```

Winsorizing - A procedure

But this still drives the error distributions, it still can affect means, coefficients, etc.

For this reason, I (tentatively) recommend the following when dealing with massive outliers in model building:

1. Split your data into testing and training sets
2. Winsorize your training data
3. Do not winsorize your test data.

A procedure

This two-step procedure is not brilliant. Its not perfect.

But it will:

1. Reduce the effect of outliers on coefficients and predictions
   - Thereby preventing too much overfitting
2. Without messing up our understanding of out-of-sample fit, or scale of possible prediction errors
   - Thereby making sure we don't shoot ourselves in the foot by sweeping possible large errors under the rug.

Censored Data

Often survey data will be winsorized when we get it. In particular, some values may be "top-coded" – e.g. setting all incomes in a survey that are above \$1million to \$1million.

This functions to preserve privacy of surveyed individuals. But it can also screw with the types of conclusions we can draw. We can't take that raw data and estimate a mean, or a full distribution, etc.

Censored Data

When someone has already winsorized, we can't then have "un-winsorized" test data for OOS predictions and errors.

- Best you can do is guess at the tail distribution (usually exponential), and add errors to the 'top-coded' observations in your test data.

Survey Weights

Surveys will also often try to 'upsample' some subpopulations. This makes it easier to draw firm conclusions *about* those subpopulations – by increasing the sample size within the subgroup.

But it means we can't just take the data at face value. If we care about interpreting coefficients, or making predictions about the general population, we need to reweight the observations to get an unbiased set of estimates.

For instance, if wealthy individuals are oversampled, such that someone with wealth over $10million is twice as likely to be sample as the median person, we want to give them half the weight in our model building – because we are half as likely to observe them *out-of-sample*.

Weighting routines

In general, if you take a vector of the relative survey probabilities p for each person, the weights that are best are usually proportional to $1/p$.

With that vector of weights, most packages will make it very easy to incorporate the weights into the model. E.g.

```
lm(y~.,data=data,weights=1/p)
rpart(y~.,data=data,weights=1/p)
ranger(y~.,data=data,case.weights=1/p)
```

Etc.

Bayes

Bayes Proof

Start with the definition of joint probability.

$$P[\beta \cap X] = P[\beta|X]P[X]$$

We know the joint probability is symmetric.

$$P[\beta \cap X] = P[X \cap \beta]$$

Combine these two facts

$$P[\beta|X]P[X] = P[X|\beta]P[\beta]$$

Bayes Rule

$$P[\beta|X] = \frac{P[X|\beta]P[\beta]}{P[X]}$$

Given some observed data or event, X, $P[X]$ is constant across all the models we want to consider. So you will often see Bayes rule rewritten, ignoring the "scaling parameter".

Bayes Rule

$$P[\beta|X] \propto P[X|\beta]P[\beta]$$

- $P[\beta|X]$ is referred to as the posterior
- $P[X|\beta]$ is the *likelihood* (we've seen before)
- $P[\beta]$ is the prior

Bayes Thinking

This is about formalizing uncertainty.

- How many whales are in the ocean right now?

The answer is a single number. In principle, all other numbers are wrong. But we don't know the number, and if we need to make decisions on the basis of that number, we need some sense of what values are *probable*.

Uncertainty

One perspective on probability is that it is purely about long-run frequencies of recurring events.

In that perspective, the statement, "there is a 60% chance there are more than 10,000 whales" doesn't make sense. There either are and the probability is 100% or there aren't, and the probability is 0%.

This is... unhelpful. In settings where decisions need to be made, getting some sense of the scale of uncertainty is critical. Using probability for that *makes sense*.

Bayes

Why does this matter?

We've spoken about how to choose between models a bit. We use data to determine performance, and pick the best one.

We've spoken about how to choose weights in ensemble models a bit.

We've spoken about how biasing coefficients towards 0 can improve out of sample performance.

All of this is fairly *strange* in a frequentist world, and all of it is very easily justified in a Bayesian world.

Why?

Adding a prior often makes these decisions sensible.

If we think most coefficients are probably 0, then incorporating that belief into our modeling will improve things.

If we think some models are likely better than others, incorporating that will improve things.

Primary Sources of Uncertainty in Modeling

1. Within the model, there is uncertainty.
   - What are parameter values
   - What value will an observation take
2. We are not certain if the model is correct
   - Should we use a different model
   - Should we average with a different model
3. We don't know if the data is correct
   - Is there a clerical error?
   - Is there measurement error?
     - E.g. Covid Case counts in the US last March – woefully undercounted.

Bayes

Bayes is primarily a computational field. Solving the problem of "what is the best coefficient" etc, is often analytically impossible, but relatively straightforward (if slow) computationally.

If there is time, we may revisit this in week 9. But broadly, this is an incredibly important subfield which languished for centuries because of computational issues that we've more or less overcome in the last two decades.

Most of the way I've taught this course, most of my intuition around statistics, and more derives from Bayes.

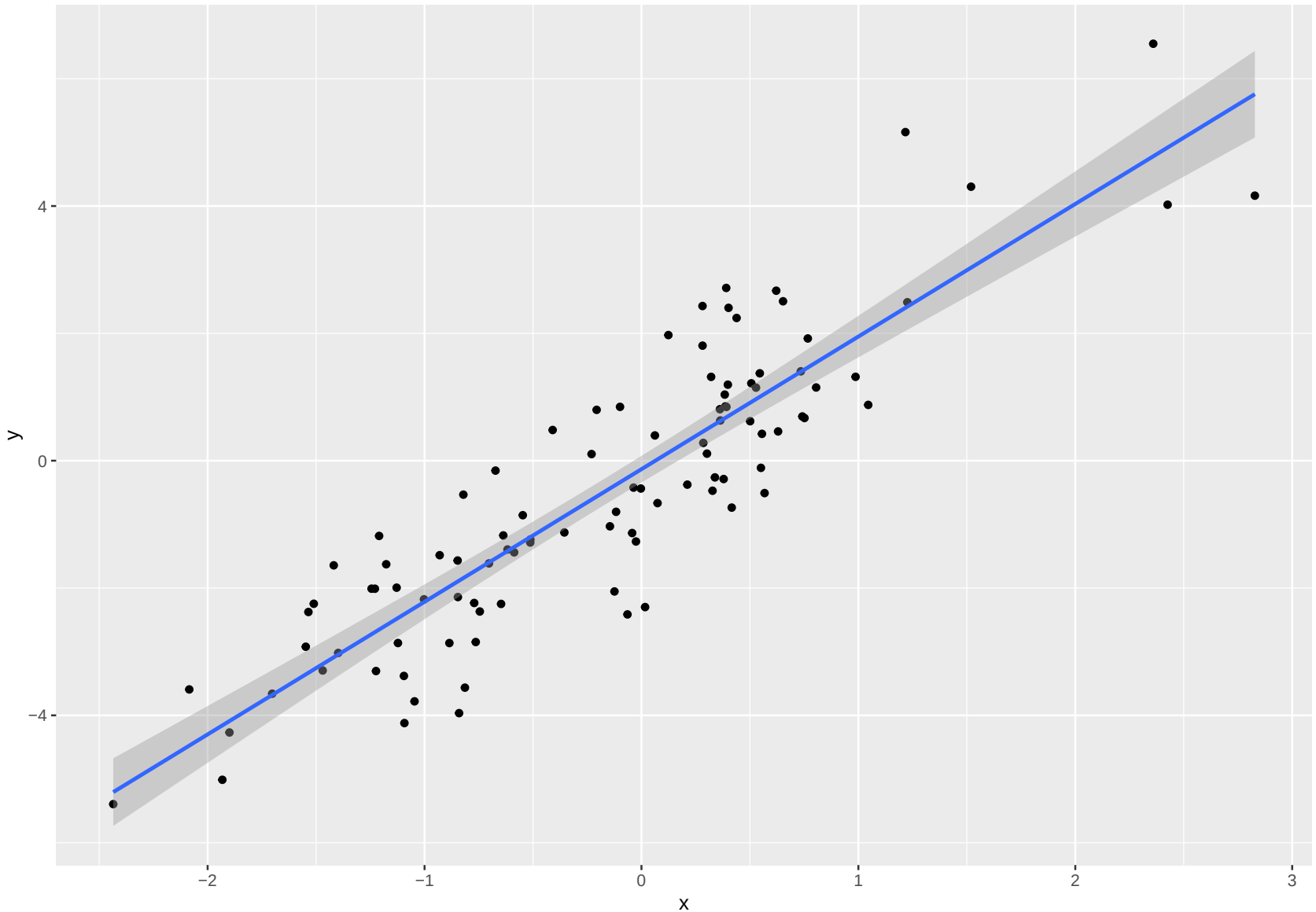
Predictions 2

Updates:

"Result" continues to change. Was 24080, is currently: 27665.

CI vs PI

```
n = 100;      x = rnorm(n);   y = rnorm(n)+2*x
```



Model Uncertainty

What if we build one model, and it says $P[event] = 0$, what do I think $P[event]$ is?

- How much do you trust the model?
  - Remember our sources of uncertainty.

If a guy at a casino tells you he has a system, and you should bet it all on red for *certain* winnings, are you now certain of winnings if you bet it all on red?

- The model is not fundamentally better. You may like it more, but its not smarter than the guy at the casino.

Model Uncertainty

Why does this matter?

Models in big data settings can become reasonably certain about events not happening, or happening. They have a lot of data, they have a model, the two combine for a lot of certainty.

But the model being certain does not mean *you* need to be certain. *The data wants to trick you*

And in competitive settings, like a casino, or the stock market, "the data wants to trick you" is less a metaphor than you might think.

Model Uncertainty

This is my way of saying, while the case count is above 25k, and likely to remain there,

I think $P[> 25k] = 0$ was overconfident unless you're certain your model was right.

- The model is just a tool.

HW 5 Review after this

Wrap up

Things to do

Homework 6 will be posted tonight. New prediction competition will be released on Monday – purely optional.

See you Tuesday.

Bye!