Loss Given Default as a Function of the Default Rate

GARP

Chicago, 25 April 2013

Jon Frye

Jon.Frye@chi.frb.org

Senior Economist

Federal Reserve Bank of Chicago

Topics tonight

Set up the problem (3 slides)

The LGD function is easy and simple (2 slides)

It works well on data we've had (3 slides)

It is apt to work well on data we'll get (7 slides)

Conclusion

Credit Loss = DR * LGD

Results for a portfolio containing 10 loans

Loan #:	1	2	3	4	5	6	7	8	9	10
Exposure:	$10	$10	$10	$10	$10	$10	$10	$10	$10	$10
Default?:	0	0	0	1	0	1	0	0	1	0
Loss:	0	0	0	$1	0	$2	0	0	$3	0

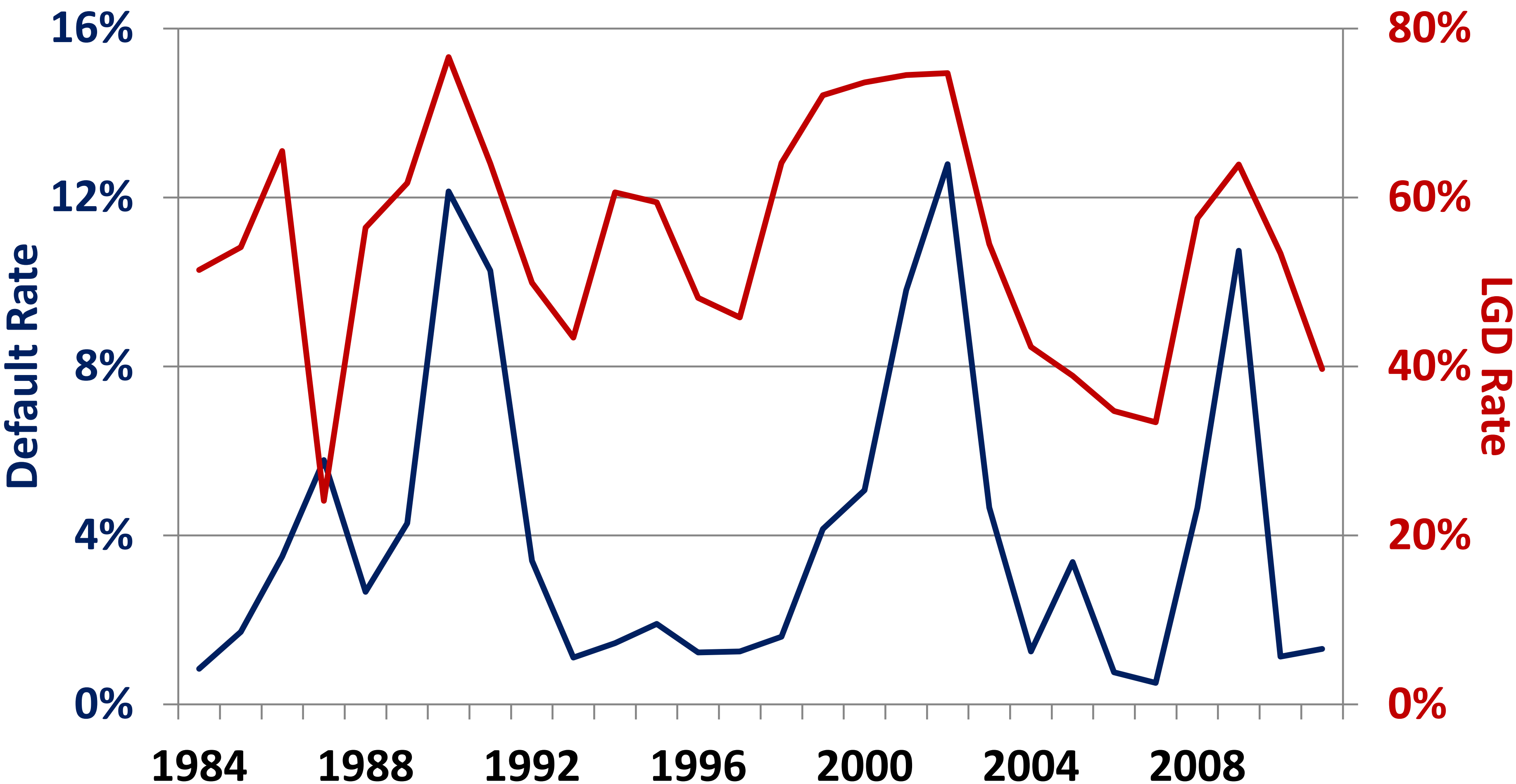
Note: All the exposure amounts are the same.

Default Rate = # Defaults / # Loans = 3 / 10 = 30%

LGD rate = Total Loss / #Defaults = (.1 +.2 +.3) / 3 = 20%

Loss rate = $6 / $100 = 6% = DR * LGD = 30% * 20%

DR and LGD move together

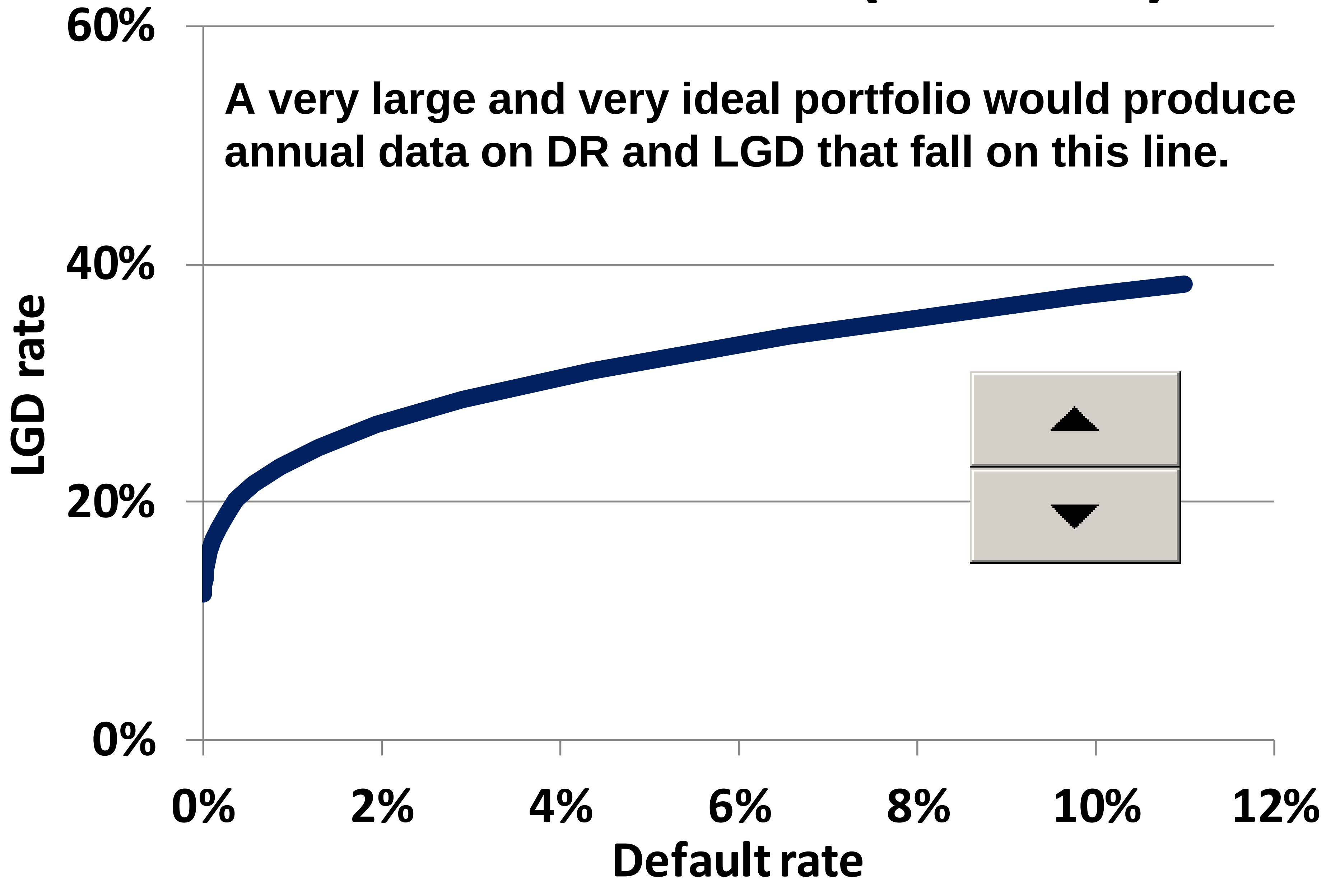


Source: Altman-Kuehne High-Yield Bond Default and Return Report, February 2012

It matters, and you might care

- **Economic capital**
  - **If LGD responds more when DR moves, there is more risk.**
- **Risk and reward**
  - **If the proportional rise of LGD is greater than average for some loans, you want more reward to make those loans.**
- **Pricing**
  - **Loans in a portfolio (or tranches of a securitization) are exposed to loss in different economic conditions.**
    - **So, their LGDs will be different, and this should be reflected in prices.**
- **Let's say you want this in a model of risk or pricing.**
  - **If DR moves, *how much* does LGD move?**

LGD function of DR (k = 0.50)



So, it is easy and simple

- To use the LGD function, just pick a line.
  - If you expect ***high*** LGD, pick one of the ***higher*** lines.
  - If you have annual data, pick a line above half the points.
    - If you want precision, use the formula for k in the paper.
  - You do ***not*** need to do statistics, just some averaging.

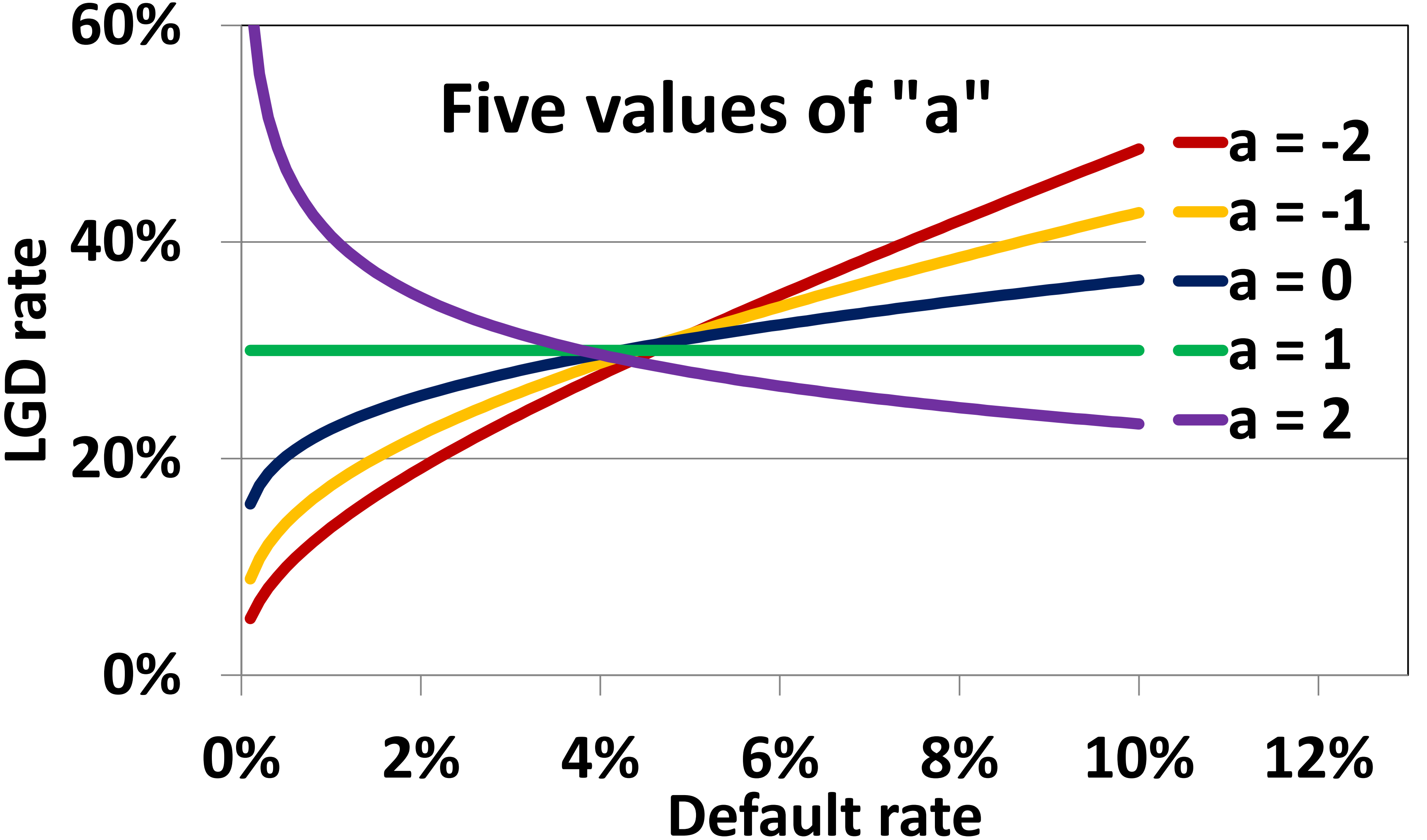
- Besides being easy to use, the LGD function implies a particularly simple model of credit loss.
  - A simpler model of LGD (e.g., linear) makes a more complicated model of loss—and loss is what matters.
    - The details are in *Journal of Credit Risk*, Spring 2012.

And, it works

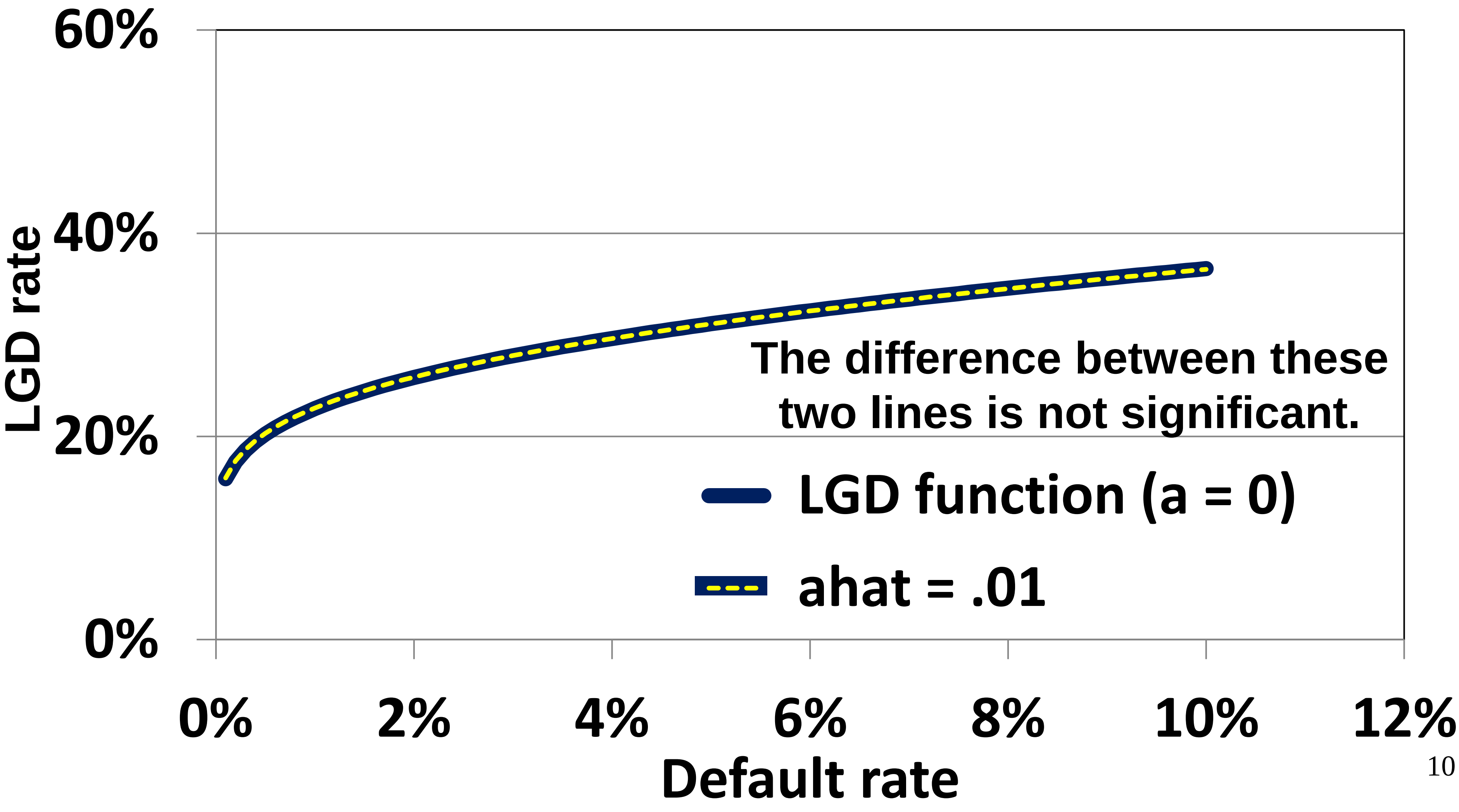
- Usually, a paper shows statistical significance.
- But, the LGD function has no statistical estimates!
- So Mike Jacobs and I worked the other way around:
  - We invented an LGD function with an extra parameter, "a".
  - We showed that "a" controls the slope of the relationship.
  - We found that "a" is ***not*** statistically significant.
    - The LGD function without "a" has the right slope to describe the data.
    - It seem to be an adequate description.

Source: JCR Spring 2012, Frye and Jacobs, *Credit Loss and Systematic LGD*

Parameter *a* controls "slope"



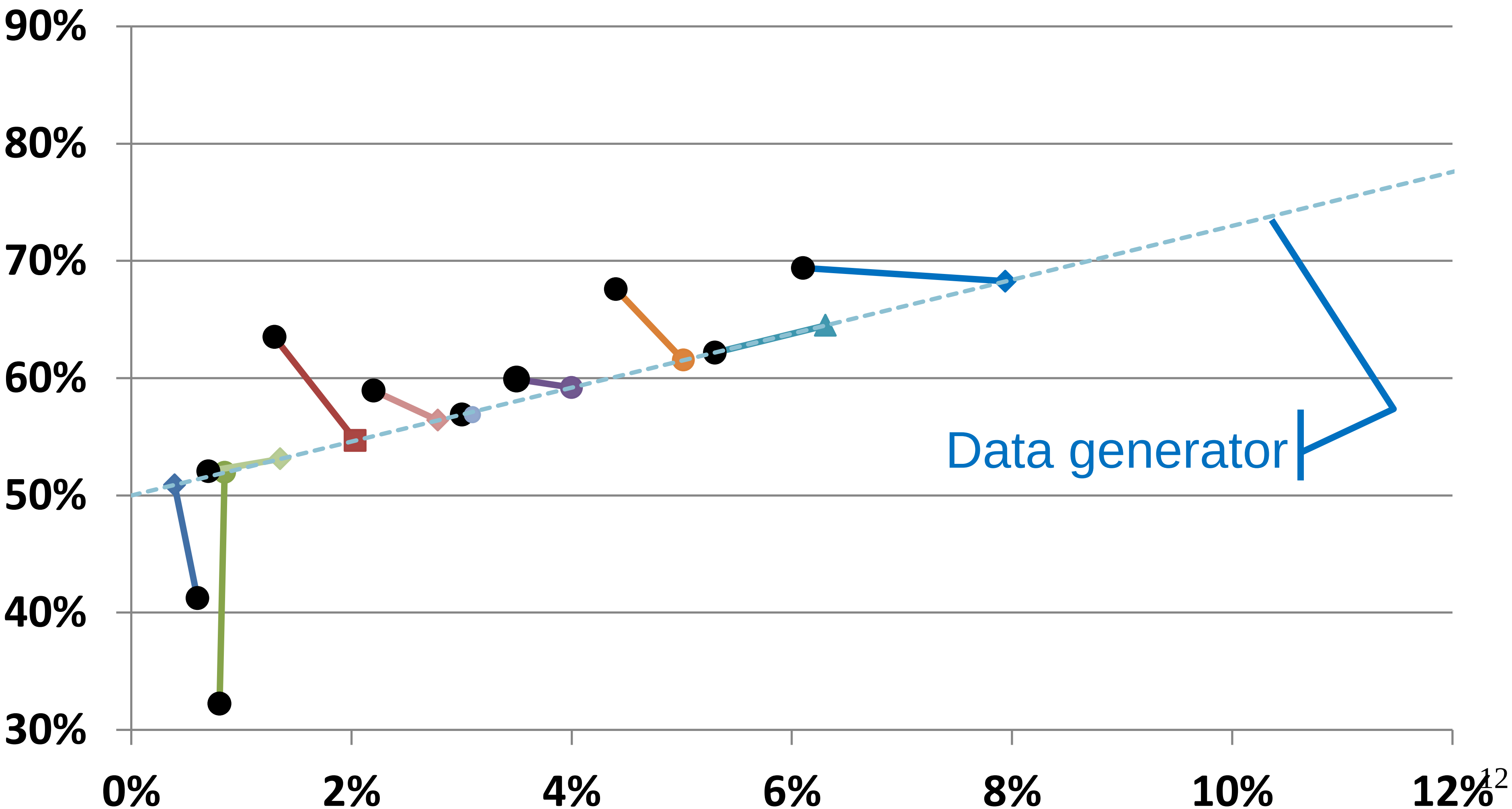
a calibrated to 14 years of loans rated Ba3, B1, B2, B3, or lower



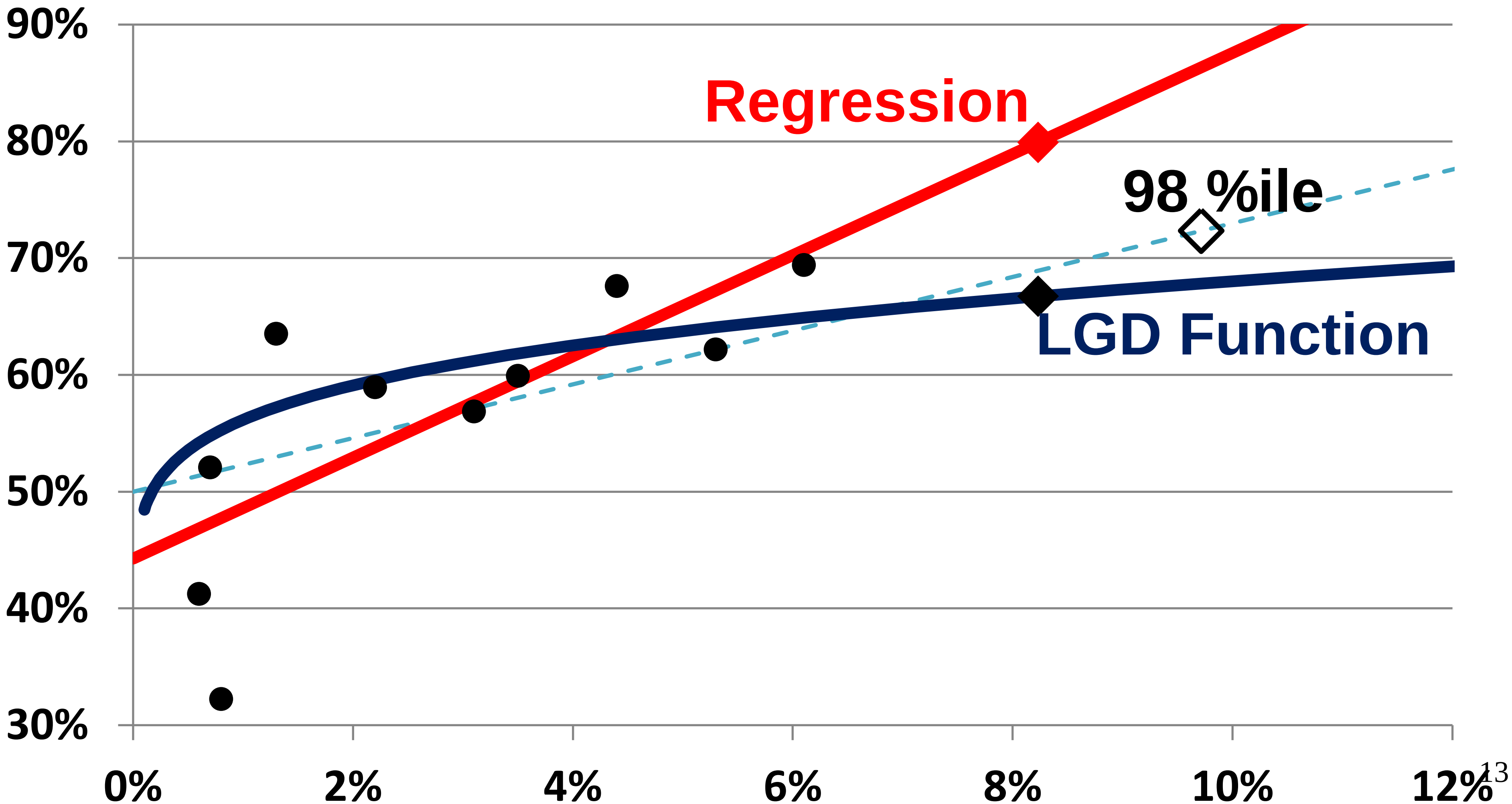
It is apt to keep working

- **The remainder of the talk draws from unpublished work.**
  - **"LGD as a function of the default rate," Frye, 2013.**
  - **http://www.chicagofed.org/webpages/people/frye_jon.cfm#**
- **It simulates ideal data of high quality.**
  - **The data we will actually have in the future won't be this good.**
- **Then it compares two forecasts:**
  - **Using a regression line drawn through the data swarm, versus**
  - **Using the LGD function.**
- **The LGD function wins, because:**
  - **The data sample is *short*.**
    - **Most banks did not define "default" or measure LGD before Basel II.**
  - **There is unavoidable *noise* in default and LGD data…**

One run: 10 years, 1,000 loans



Two forecasts

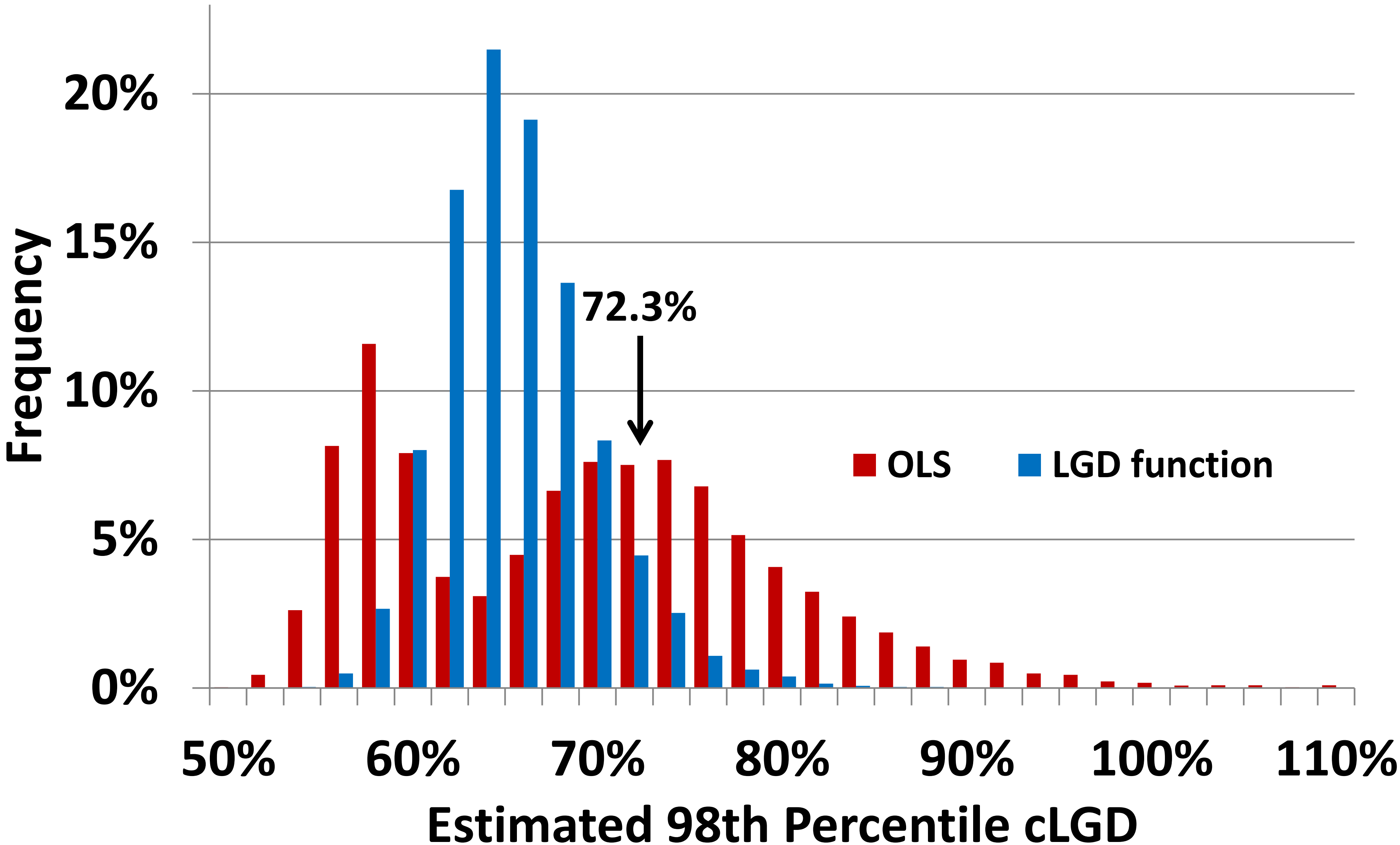


Let's review what just happened

- **The LGD function performs better, even though:**
  - **I give an advantage to *<u>linear</u>* regression by simulating the data with a *<u>linear</u>* model.**
  - **I use a data generator that is quite a lot steeper than the LGD function.**
  - **Regression actually *<u>looks</u>* *<u>at</u>* *<u>the</u>* *<u>data</u>* *<u>swarm</u>*.**
    - **LGD function calibrates only to averages, not to the slope.**

- **So far, though, I've only shown it works better for one particular set of simulated data.**
  - **The next slide looks at 10,000 sets of data simulated in exactly the same way…**

Tally 10,000 runs



Other simulation experiments

- **A skeptic wonders whether the LGD function would outperform under other conditions.**
- **I tried everything.**
  - **Different values of all simulation parameters.**
  - **Different number of loans in the portfolio.**
  - **Different number of years of data.**
- **Two things matter to the conclusion:**
  - **The number of *years* of data, and,**
  - **To a lesser extent, the *slope* of the data generator**

How to beat the LGD function

- **Get more data.**
  - **Beware: "asymptotic" qualities do not kick in next week.**
  - **You will need 20 "years" for some sets of parameter values, and you will need more than 100 "years" for some other sets.**
    - **By "years," I mean independent random draws in simulation.**
    - **If you get data from the real world you will need *more* data than this, because each year of real-world data is a partial rehash of previous years.**
    - **"If you ignore autocorrelation, you exaggerate the significance of results."**

- **Have more LGD risk.**
  - **A steep data generator tends to make steep data swarms.**
  - **The regression notices this and the LGD function doesn't.**

In this most precious meantime

- The LGD function works well:
  - It is consistent with historical data.
  - It is consistent with a super simple credit loss model.
  - It is easy to apply.
  - It assigns some LGD risk to every credit exposure.
  - It survives statistical tests using historical data.
  - It is likely to outperform statistical analysis for a long time.

Questions?

Thank you for your attention