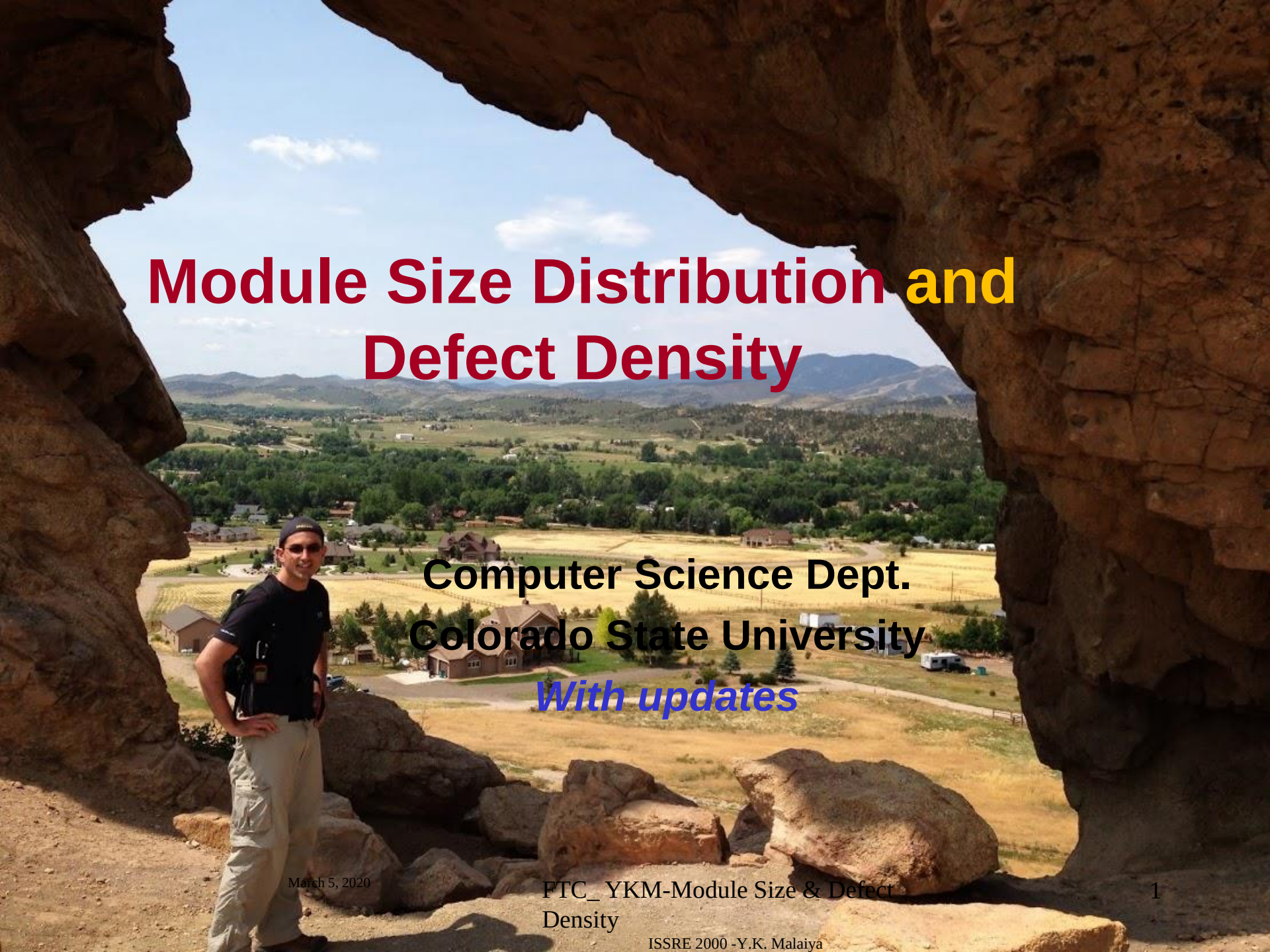


Module Size Distribution and Defect Density

Computer Science Dept.

Colorado State University

With updates

Module Size Distribution and Defect Density

- **Significance: Factors that affect defect density**
- **Existing work: Data & Hypothesis**
- **A New Composite Defect Density Model**
- **Available Data & Model**
- **Module Size Distribution: Predictable?**
- **Total Defect Content & Implications**
- **Observations & Conclusions**

Factors Affecting Defect Density

- **Multiplicative models:**
  - **RADC**
  - **ROBUST**
- **Sub-models:**
  - **Phase**
  - **Programming team**
  - **Process maturity**
  - **Structure**
  - **Requirement volatility**

Earlier Studies

- **Shen et al.: For modules > 500 lines, no size-density relation. Smaller modules: density declines with size**
- **Banker and Kemerer: Hypothesis for optimal module size**
- **Withrow: minimum near size ≈ 200**
- **Hatton: two separate models for smaller & larger modules**
- **Rosenberg: module size-defect density correlation misleading**
- **Fenton and Ohlsson: no significant dependence observed**

A Composite Defect Density Model

- **Module-related faults: associated with**
  - **parameters passed among the modules,**
  - **assumptions made by modules regarding each other,**
  - **handling of global data,**
  - **Assumption: such faults are uniformly distributed among the modules.**
- **Instruction-related faults: *bulk* defect density. Assumption: defect density components are**
  - **constant,**
  - **number of other instructions a given instruction may interact with.**

A Composite Defect Density Model

- Module related defect density :

  total defects/mo dule : a, module size : s

$$D_m(s) = \frac{a}{s}$$

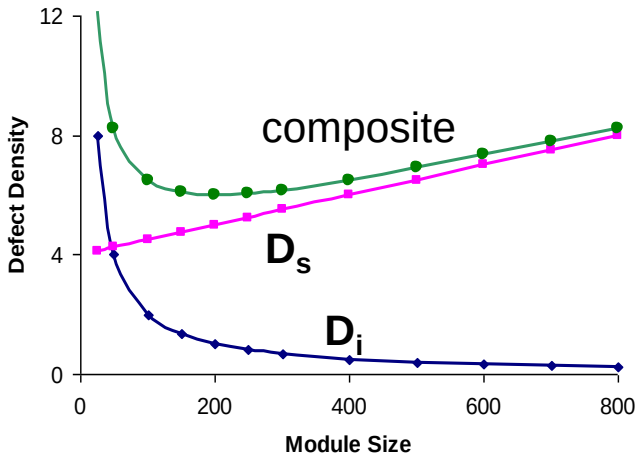
- Instructio n related defect density

$$D_i(s) = b + cs$$

- The composite defect density is then

$$D(s) = D_m(s) + D_i(s)$$

$$= \frac{a}{s} + b + cs$$



The Two Regions

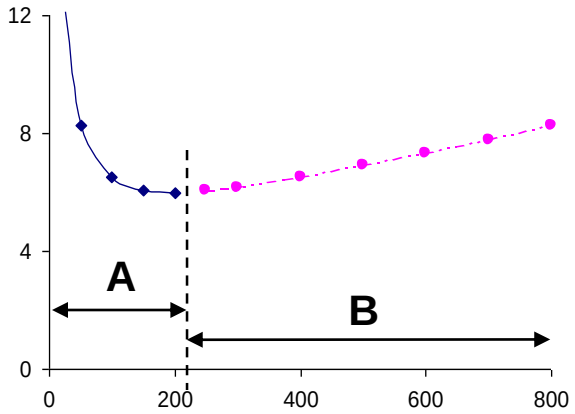
- The minimum defect density $D_{\min} = (2\sqrt{ac} + b)$

  occurs at module size $s_{\min} = \sqrt{\frac{a}{c}}$

- Model implies two regions:

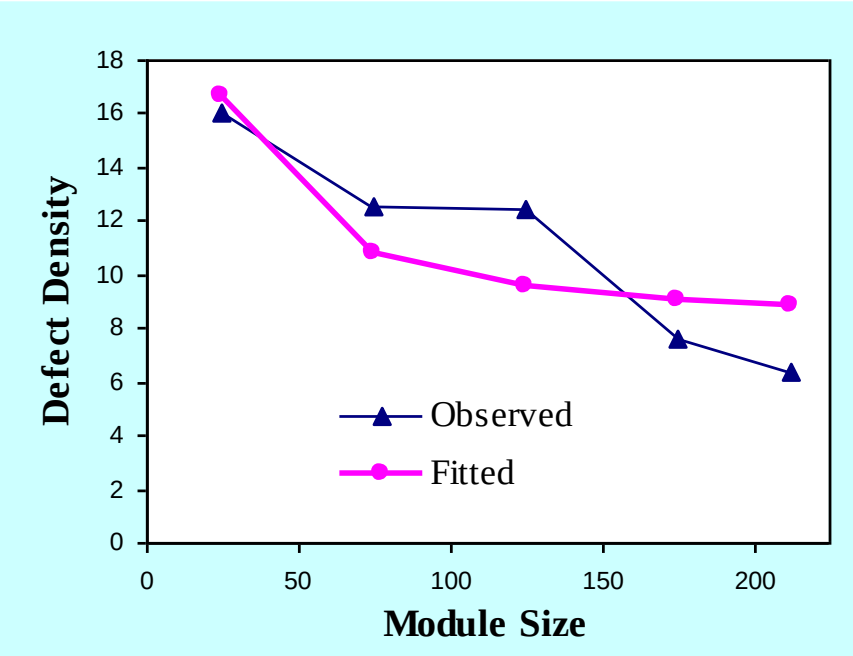
  Region A : For modules with $s < s_{\min}$

  Region B : For modules with $s > s_{\min}$



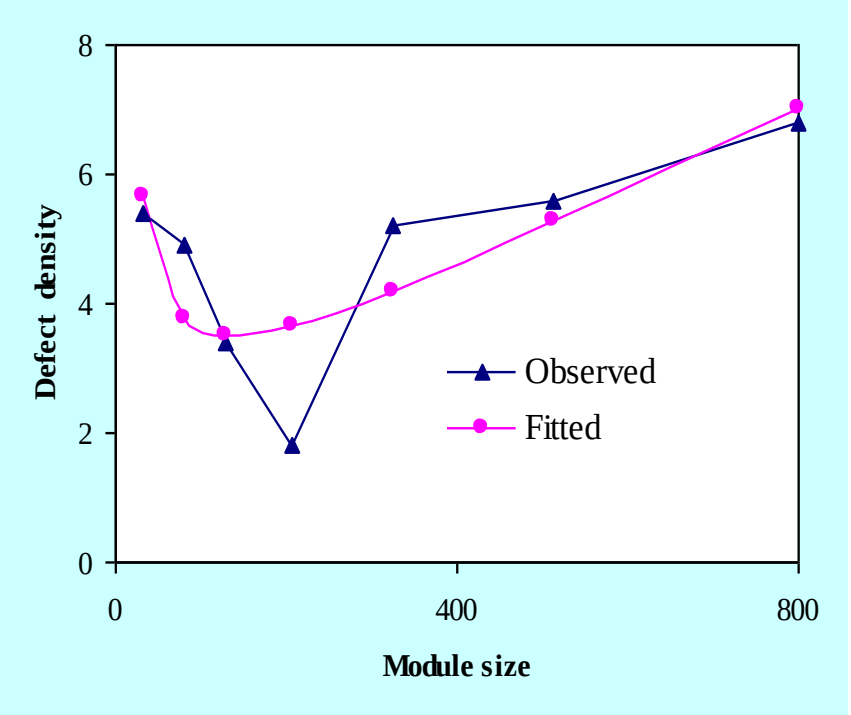
Data: Basili & Perricone

Module Size (max)	Module count	Cyclomatic Complexity	Defect Density (/KLOC)
50	258	6	16
100	70	17.9	12.6
150	26	28.1	12.4
200	13	52.7	7.6
225	3	60	6.4

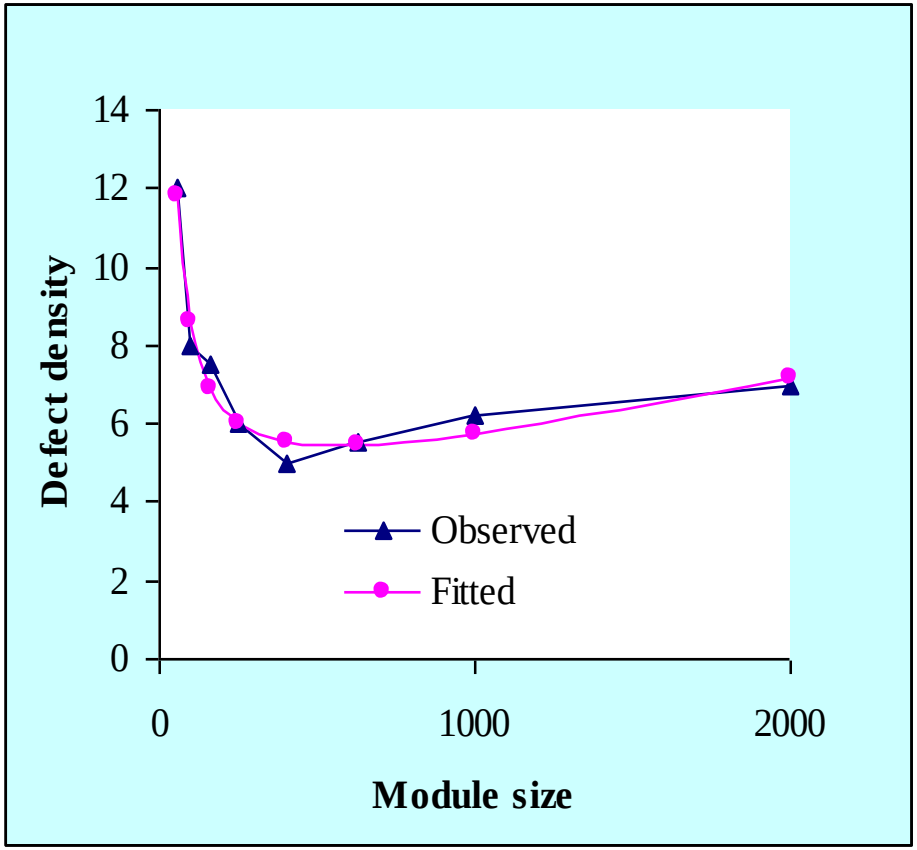


Withrow Data

Source Lines	Modules	Defect Density
4-62	93	5.4
64-97	39	4.9
103-154	52	3.4
161-250	53	1.8
251-397	46	5.2
402-625	31	5.6
651-949	22	6.8
1050-5160	26	8.3



Columbus Data



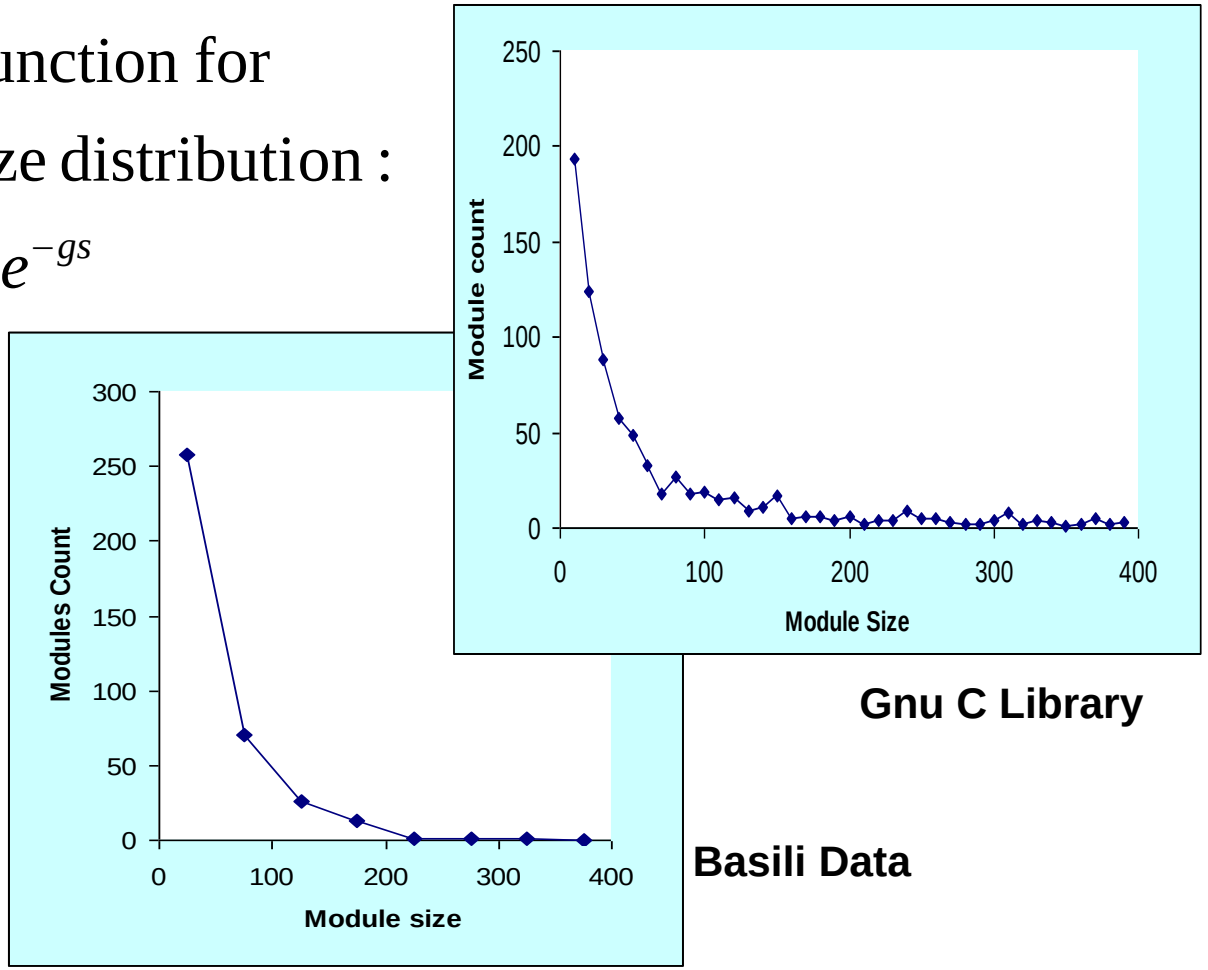
Parameter Values

Data	S_{min}	Parameter Values		
		a	b	c
Basili	NA	220.9	7.83	0
Columbus	400	223.79	4.73	0.0013
Withrow	200	121.19	1.76	0.0063

Distribution of Module Sizes

- Density function for module size distribution :

$$f_s(s) = g.e^{-gs}$$



Gnu C Library

Basili Data

Module Size Distribution: Parameters

Data	Language	M (total modules)	Parameter g
Basili	Fortran	370	0.0054
Withrow	ADA	362	0.0041
Shen	PL/S	108	0.0029
Gnu C Lib	C	792	0.0097

Overall Defect Density

- If S_T : total project size, M : number of modules, the overall defect density is

$$D = \frac{\int_1^{s_{\max}} Mge^{-gs}(\frac{a}{s} + b + cs).10^{-3}.s.ds}{S_T}$$

- Example : M = 400, g = 0.004, largest module 2000 lines, a = 120, b = 1.8, c = 0.006

$$s_{\min} = 141.42$$

$$S_T = \int_1^{s_{\max}} Mge^{-gs}.s.ds = 100{,}000 \quad lines$$

D = 7.09 per KLOC

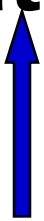
Optimal Module Size Distribution?

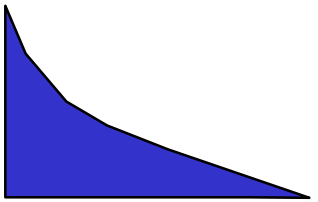
- If all modules can be equal, make them $s_{\min}$.
- If they are exponentially distributed :

$$D = \int_{1}^{s_{\max}} g^2 e^{-gs} \left(\frac{a}{s} + b + cs\right).10^{-3}.s.ds$$

$$\approx 0.001\left(ag + b + 2\frac{c}{g}\right)$$

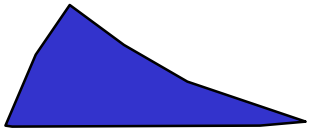
hence $g_{opt} = \sqrt{\frac{2c}{a}}$ and $s_{opt} = \sqrt{\frac{a}{2c}} = \frac{s_{\min}}{\sqrt{2}}$





- Merge smaller modules resulting in a peak near $s_{\min}$.

Merge smaller



Sub-model: Module Size Distribution

- Multiplicative sub-model
- Default value: 1
- Parameters estimated using calibration
- Assuming exponential distribution

$$F_{ms} = Ag + B + \frac{C}{g}$$

- Example: If $a = 120, b = 1.8, c = 0.006,$ and default $g = 0.005$

$$F_{ms} = 25g + 0.375 + \frac{2.5 \times 10^{-3}}{g}$$

Observations on Data

- **Trends not observable if number of modules is small.**
- **Trend for region B not observed if size<s_{min} for most modules, as in Basili & Perricone's data (very few modules >400). Weak dependence.**
- **Trend for region A not observed if size>s_{min} for most modules, as in Fenton and Ohlsson's data (very few modules with size <500). Stronger dependence.**
- **Selective testing or uneven reuse may mask dependence.**
- **Avoiding very small modules may be more beneficial than avoiding very large modules.**

Conclusions

- **A model explaining both declining and rising defect density trends.**
- **Module size distribution is often exponential due to natural reasons.**
- **A defect density model to take variation in size distribution into account .**
- **Adjusting size distribution may minimize defects.**
- **Impact of merging or breaking modules needs to be studied.**

Recent Developments

Rosenberg's Analysis

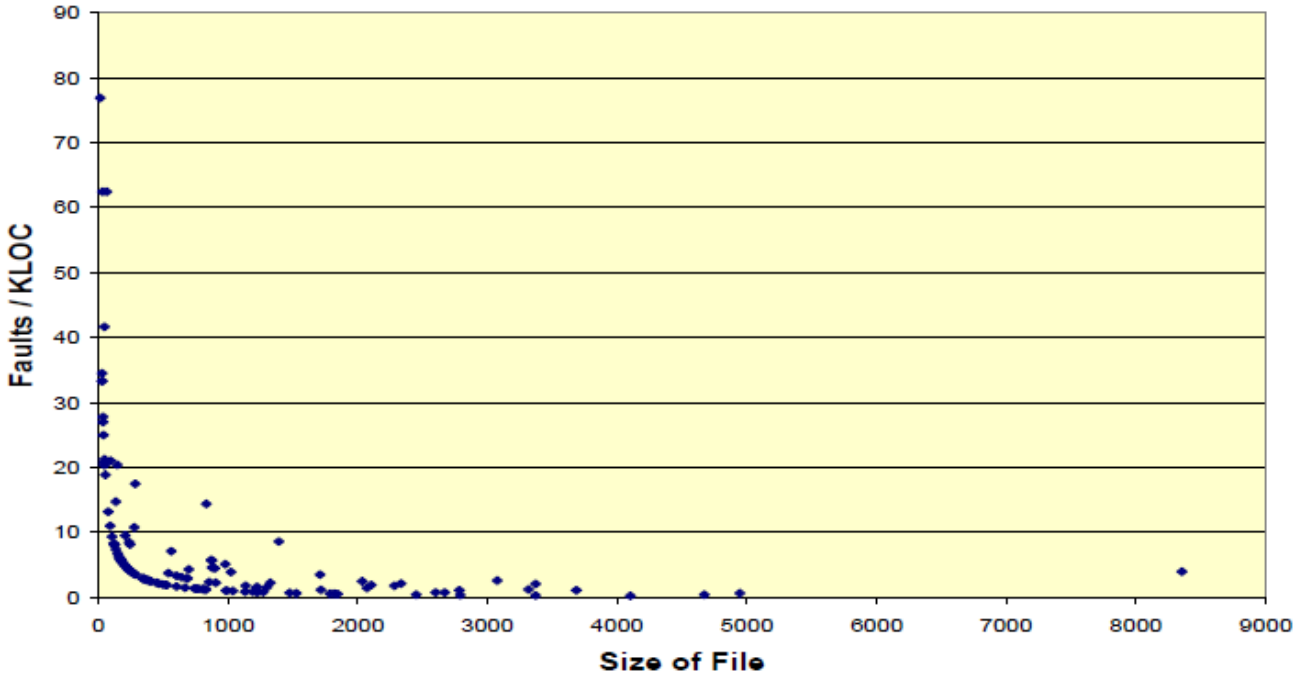
- **Argument: If we assume X and Y are statistically independent.**
  - **Then scatter-plot of Y/X against X looks like *declining defect density vs. module size plot* (Region A).**
- **Flaw: Note that assumption implies that total defects in a module is independent of module size, i.e. defect density is inversely proportional to module size.**

J. Rosenberg, "Some misconceptions about lines of code," Proc. Int. Software Metrics Symp, pp. 137-142, Nov. 1997.

A student wrote in 2011

- **I almost caused a riot at work when I mentioned that there was data showing that larger software modules had a lower defect density that more smaller modules.**

AT&T Study



- **Thomas J. Ostrand and Elaine J. Weyuker. 2002. The distribution of faults in a large industrial software system. In Proceedings of the 2002 ACM SIGSOFT international symposium on Software testing and analysis (ISSTA '02). ACM, New York, NY, USA, 55-64.**

Fenton & Ohlsson: no modules < 500

Table 4. Faults/1000 Lines of code release n and n+1.

	Release n		Release n+1	
Module size	Frequency	Faults/1000 Lines	Frequency	Faults/1000 Lines
500	3	1.45	6	13
1000	15	4.77	17	6
1500	32	5.24	35	5
2000	24	6.32	41	7
2500	14	5.88	34	5
3000	22	5.74	37	5
3500	11	7.83	18	7
>3500	9	7.38	42	8

Fenton, N., Ohlsson, N.: Quantitative analysis of faults and failures in a complex software system. IEEE Transactions on Software Engineering, 797–814 (2000)

Koru et al.

- **In Mozilla and Eclipse, an inspection strategy investing 80 percent of available resources on 100-LOC classes and the rest on 1,000-LOC classes would be more than twice as cost-effective as the opposite strategy.**
- **We observed that defect proneness increased with module size but at a smaller rate. Therefore, smaller modules were proportionally more defect prone compared to larger ones.**

A. G.; Koru, D. Zhang, K. El Emam, Hongfang Liu, "An Investigation into the Functional Form of the Size-Defect Relationship for Software Modules," IEEE Trans. on Software Eng., vol. 35, no. 2, pp. 293-304, March/April, 2009