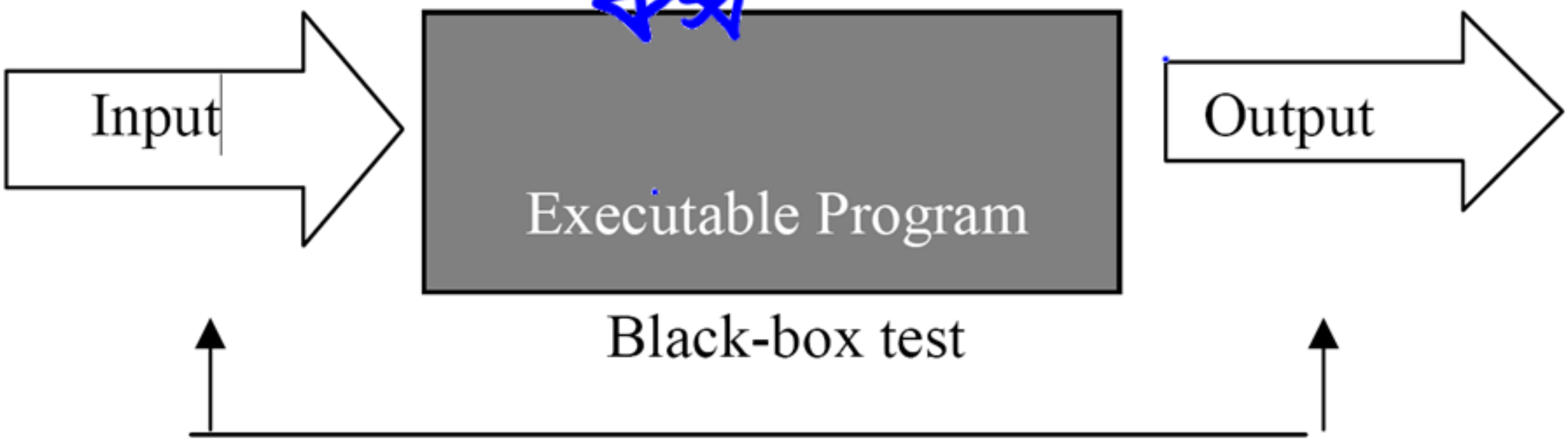
Boundary Value Testing

Lecture Objectives:

1) Define a Software Boundary condition
2) Explain why boundary value conditions represent commonly occurring mistakes
3) Given a software description, construct test cases using the boundary value testing method
4) Compare and contrast boundary value testing with equivalence class testing
5) Based on boundary value testing, determine the minimum number of tests required to test a given system
6) Construct test cases combining equivalence classes and boundary value testing to test multi-variable problems.

Good Afternoon class.... How is everyone today?

Simple Black Box Testing Model



$$y = x^3 + 6x^2 - 50x + 3$$

Boundary Conditions

$y' = 3x^2 + 12x - 50$

2.546,
-6.546

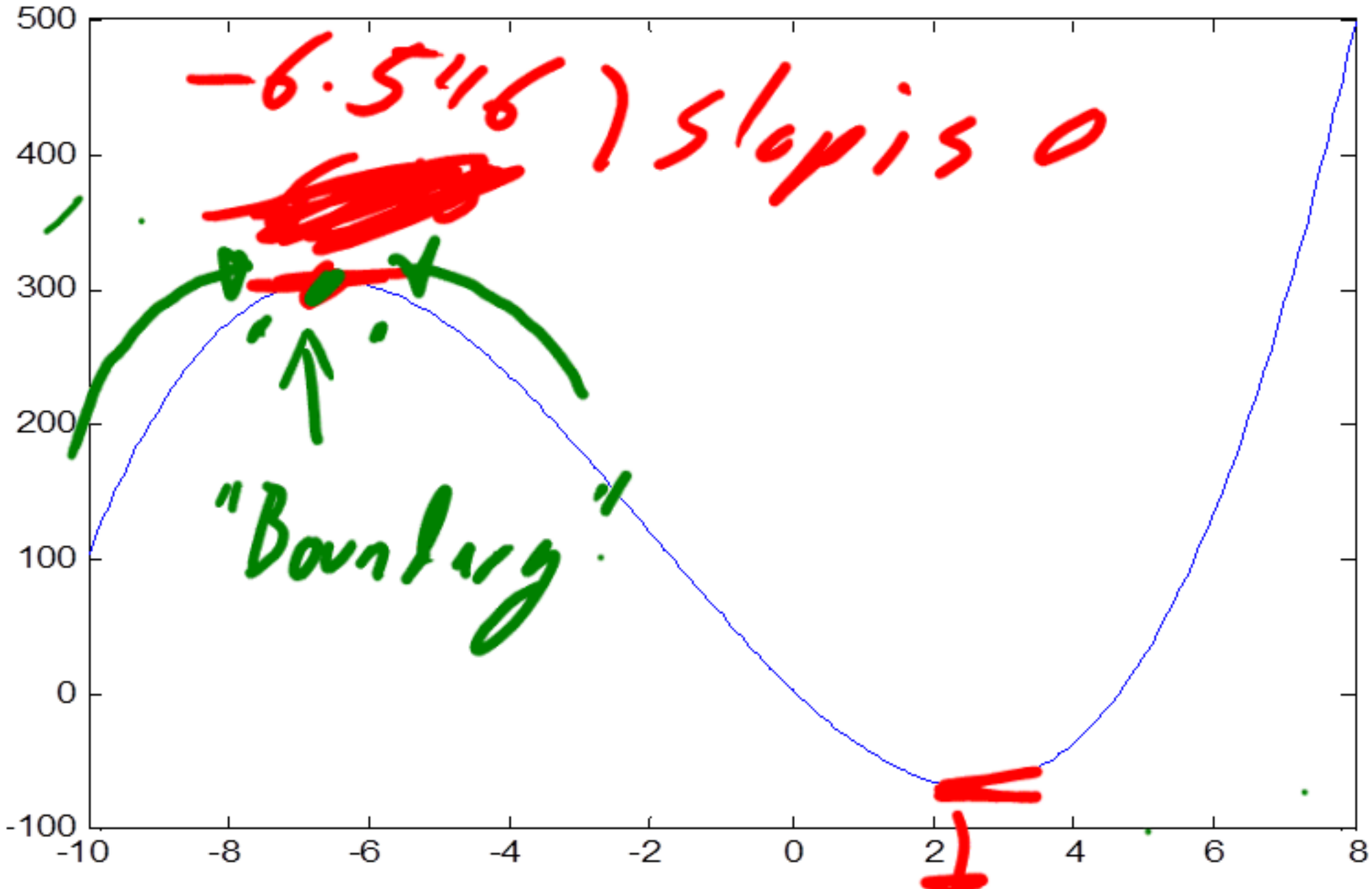
Zeros of derivative

– critical locations

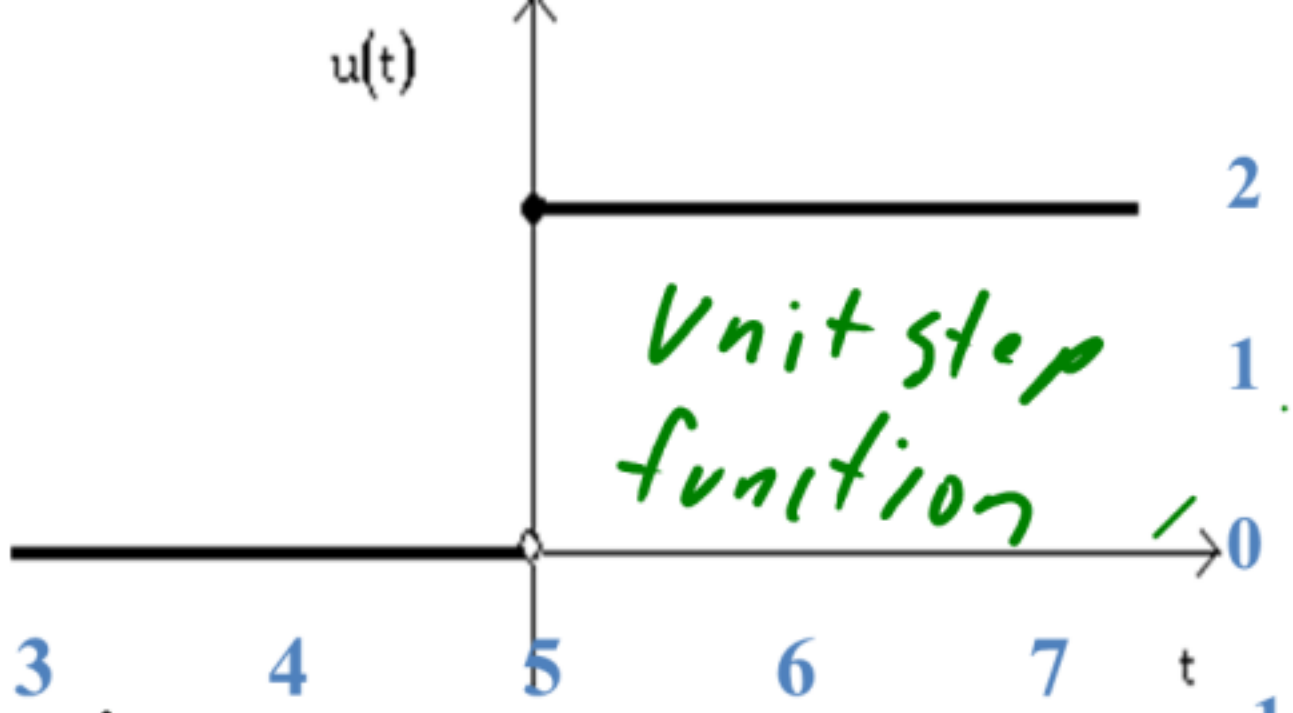
Boundary Conditions

$$y = x^3 + 6x^2 - 50x + 3$$

$$y' = 3x^2 + 12x - 50$$



Boundary Conditions



4 ⇒ 0

5 ⇒ ≠2

6 ⇒ 2

Code:
```
if (x<5)
{ y=0; }
else { y=2; }
```

Boundary Value Analysis

- Boundary value analysis(BVA) - a test case design technique
  - complements to equivalence partition
- Objective:
  - Boundary value analysis leads to a selection of test cases that exercise bounding values.
- Guidelines:
  - If an input condition specifies a range bounded by values a and b, test cases should be designed with value a and b, just above and below a and b.
- Example: Integer D with input condition [-3, 10]
  - test values: -4, -3, -2, 9, 10, 11

Equivalence Class Example

- Math.absoluteValue(int val)

Boundary Value:

-1 + one below boundary

0 ← at boundary

1 ← "Above" boundary

Boundary Value Analysis

- Think of a store which offers bulk
- discounts on a purchase.
- Number of units Price per unit
  - First 10 Units $5.00
  - Next 10 Units $4.75
  - Next 10 units $4.50
  - More than 30 units at a time $4.00
- Lets write test cases for a routine
  - Double calculateTotalPrice(int quantity)

10 9 11

20 19 21

30 29 31

if(x<=10)

:

else if(x<=20)

....

Why is boundary value testing effective?

- <=, <, >, and >= are commonly problematic in implementation
  - Off by one error
- Loops are controlled by boundary conditions
- Requirements may not be understood when code is implemented

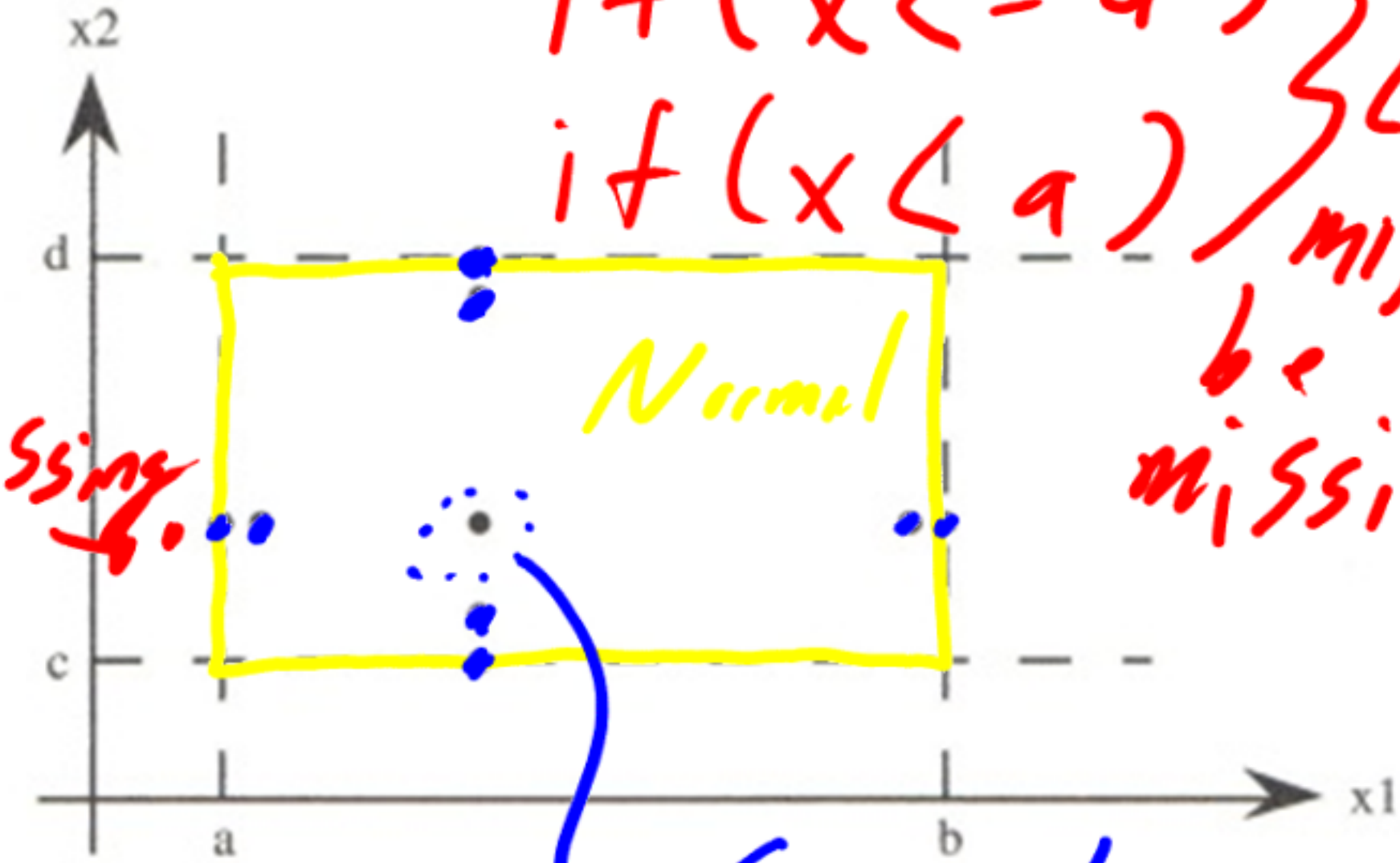
How Many Test Cases are Necessary

Boundaries (1D)	Equivalence Classes (Weak Normal)	Equivalence Classes (Strong ~~Normal~~ Robust)	Boundary Value Analysis
1	2	4	3
2	3	5	6
3	4	6	9
4	5	7	12

Four general forms

- Basic boundary value analysis
  - Tests at the boundary and one off
- Robustness testing
  - Tests at the boundary as well as one above and one below the boundary
- Worst case testing
  - More thorough than basic or robustness testing
- Robust worse case test of two variables
  - Extension of worst case testing

Basic Boundary Value Analysis 2D



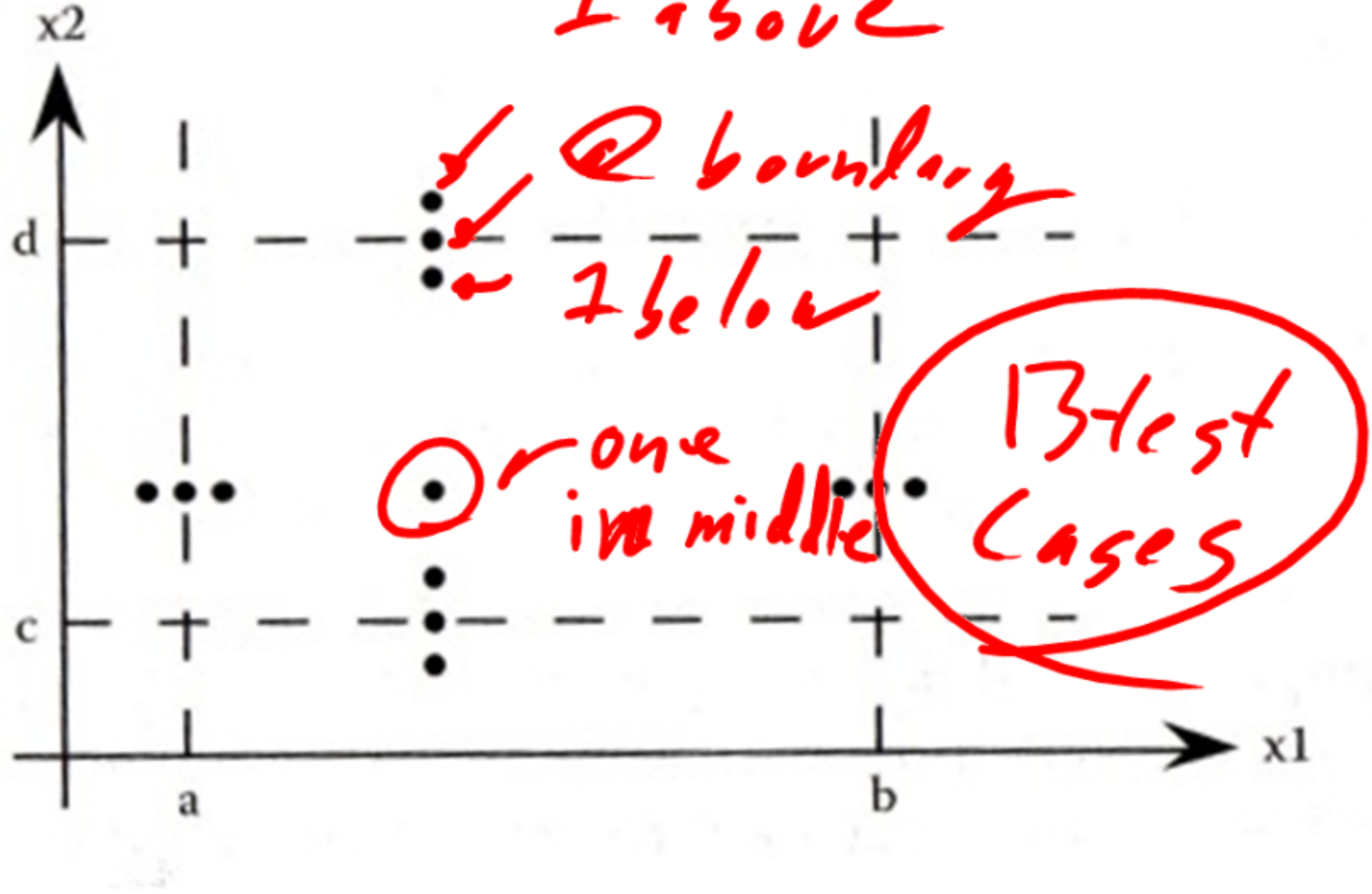
if(x <= a)
if(x < a)
} Logic might be missing

missing

Normal

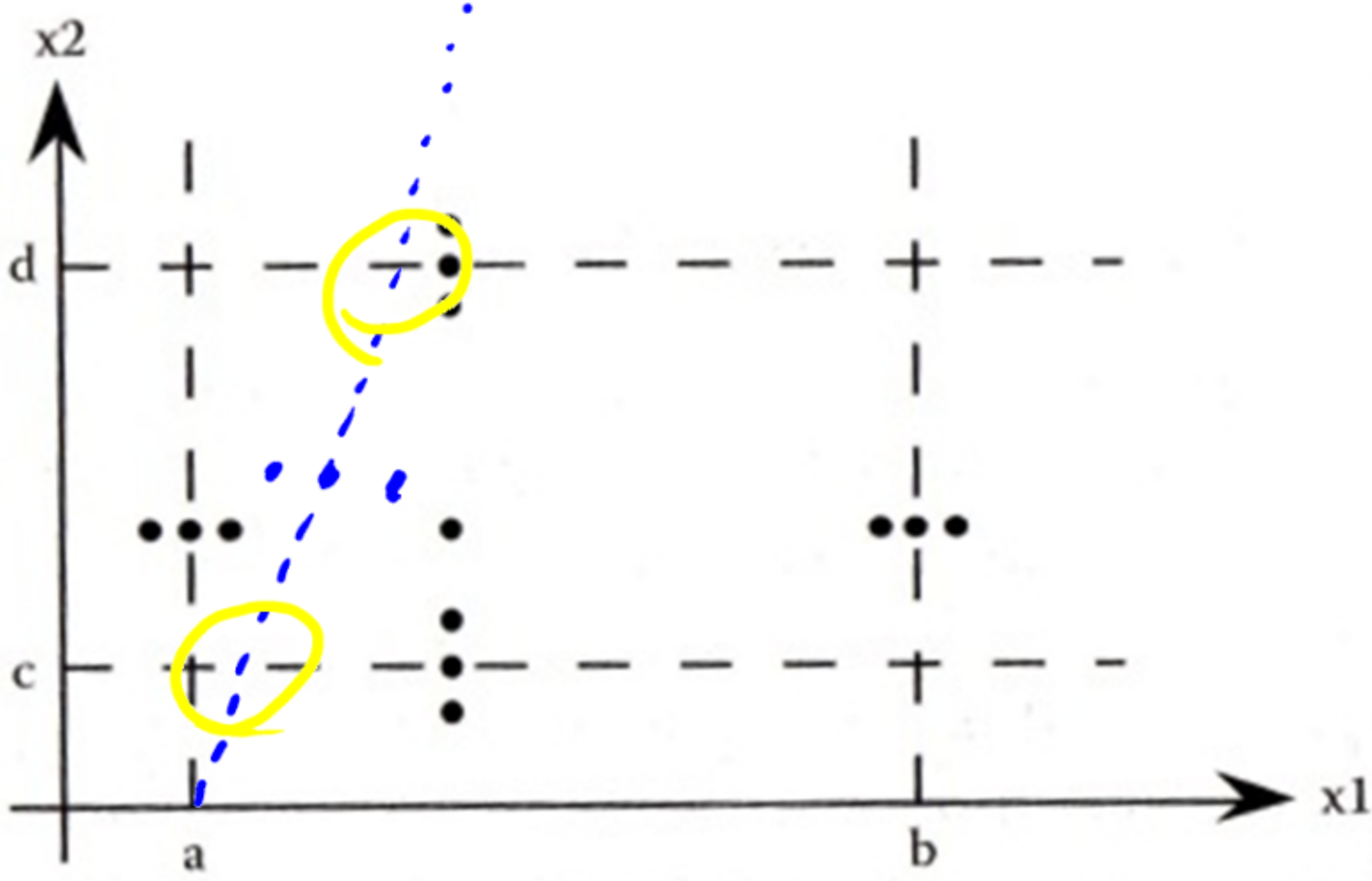
Somewhere in the middle

Robustness Testing



Above
@ boundary
below
or one in middle
13 test cases

Robustness Testing



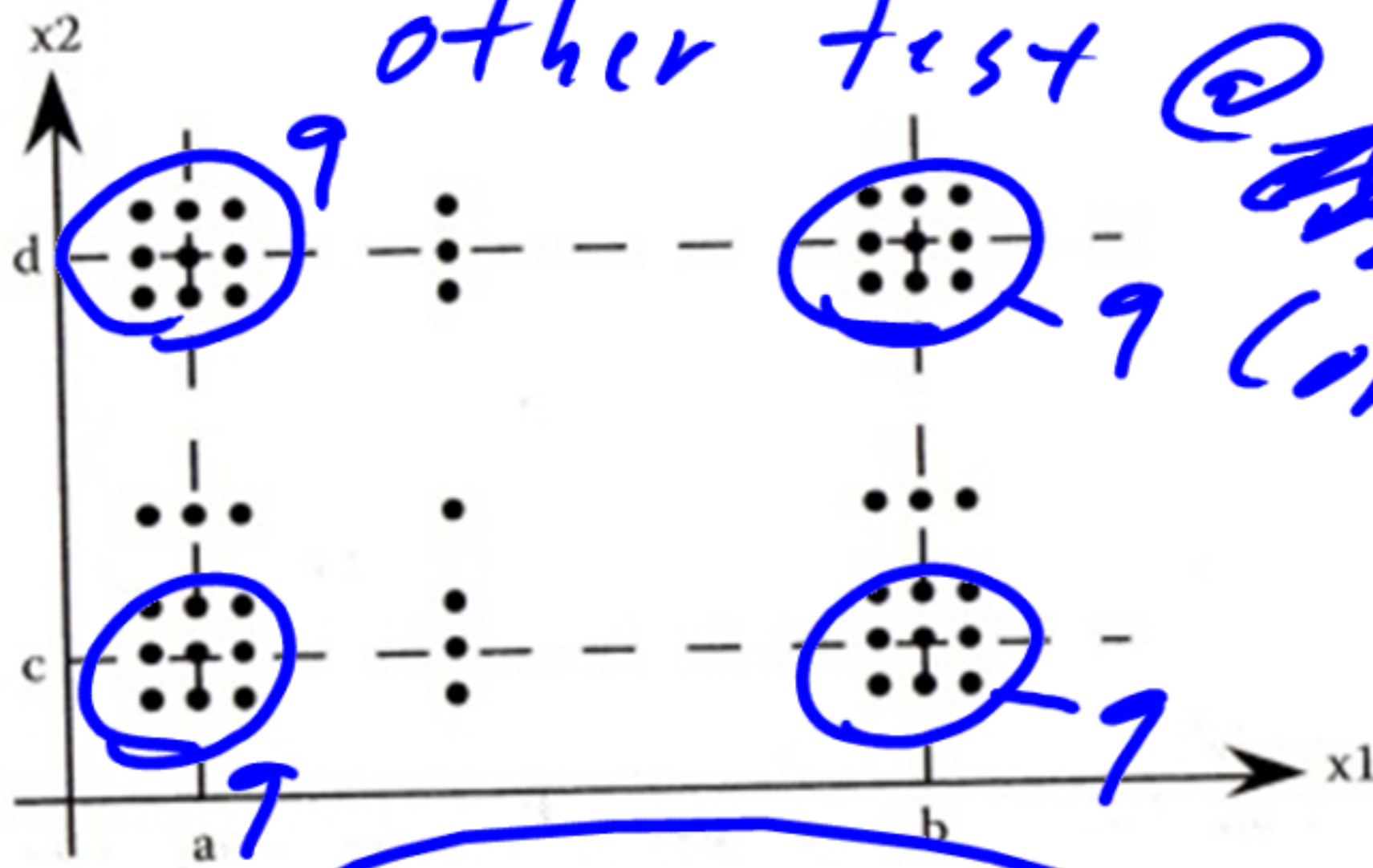
Worst Case Testing

outside is not checked

testing at all boundaries edges.

7 test 25 cases

Robust worst case testing



Robustness testing + other test @ 9 corners

9

9

7

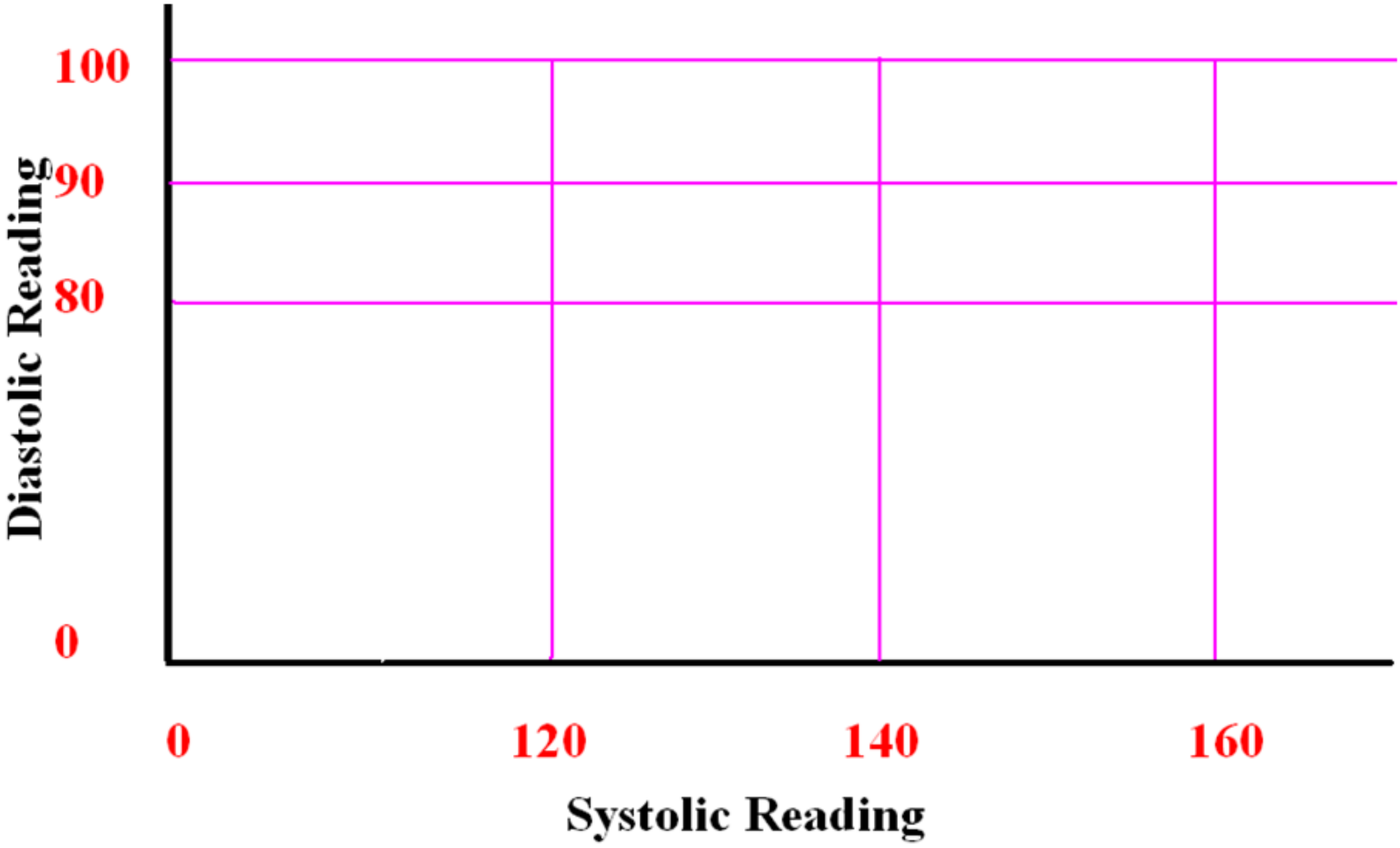
7

49 test cases

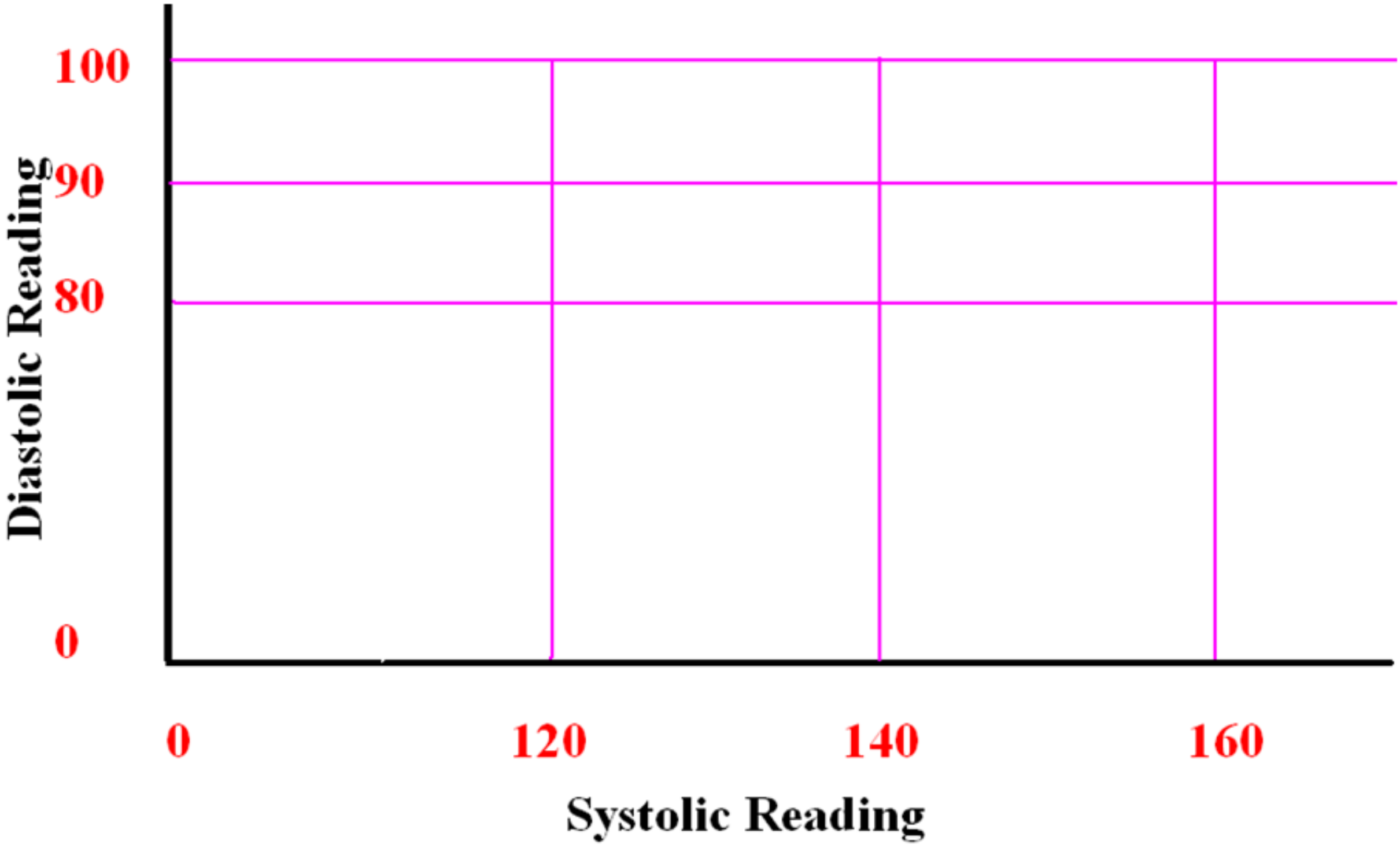
Blood Pressure Example

Category	Systolic Reading		Diastolic Reading
Normal	less than 120	and	less than 80
Prehypertension	120 – 139	or	80 – 89
High Blood Pressure (Hypertension) Stage 1	140 – 159	or	90 – 99
High Blood Pressure (Hypertension) Stage 2	160 or higher	or	100 or higher

Strong Normal Equivalence Testing



Basic Boundary Value



Robust Worst Case Boundary Value