

Metriche e Misure nell' Ingegneria del Software

Introduzione

1

1

Indice

- ◆ Metriche e Misure: teoria e pratica
 - Definizione empirica di misura
 - Definizione formale di misura (cenni)
- ◆ Sistemi di misura
 - scale e trasformazioni di scala
- ◆ Progettazione di esperimenti
- ◆ Fondamenti di analisi (statistica) dei dati
 - regressione lineare e logaritmica
 - esempi
- ◆ Come stabilire un programma di misure

2

2

Cosa è una misura (1)

- ◆ Le misure sono il cuore di molti sistemi che governano la nostra vita
 - In economia determinano il prezzo delle cose e le variazioni di prezzo
 - In medicina consentono diagnosi accurate
 - In fisica consentono la creazione di modelli del reale
 - In ingegneria (civile) sono la base per la progettazione
 - In meteorologia consentono la predizione delle condizioni atmosferiche

3

3

Cosa è una misura (2)

- ◆ Cosa hanno in comune tutte queste “misure”?
 - Il processo di fondo necessario per ottenerle

La misura è il processo attraverso il quale numeri o simboli sono assegnati a particolari proprietà di un oggetto per poterlo descrivere, secondo regole ben definite

4

4

La misura nell' ingegneria

- ◆ Le misure rappresentano un aspetto cruciale in tutte le branche dell' ingegneria...

Se esiste, lo posso misurare!

5

5

La misura nell' ingegneria del software

- ◆ Le misure rappresentano un aspetto cruciale in tutte le branche dell' ingegneria...
 - ... tranne che nell' ingegneria del software!
 - Nell' ingegneria del software si parla di proprietà come usabilità, affidabilità, e manutenibilità senza spiegare come possano essere misurate
 - Le poche misure che vengono fatte lo sono in maniera infrequente, inconsistente e incompleta, spesso senza uno scopo preciso
 - » data cemeteries [Dieter Rombach]
 - La crisi del software:
 - » “You cannot control what you cannot measure” [DeMarco]

6

6

Cosa misurare

- ◆ Processo:
 - costo di sviluppo
 - produttività
 - efficacia di metodi e tool
- ◆ Prodotto
 - difettosità
 - affidabilità
 - altre qualità ...
- ◆ Punti di vista: manageriale, tecnico, utente (marketing), ...

7

7

La teoria rappresentazionale della misura

- ◆ Fornisce un quadro di riferimento rigoroso che permette di valutare la bontà di una misura
 - Definisce i concetti di misura, scala, trasformazione di scala
- ◆ Sulla base di tale teoria è possibile analizzare le misure proposte in letteratura e proporre di nuove/migliori
- ◆ Es. la misura della temperatura
 - legata a fenomeni fisici precisi come il congelamento e l'ebollizione dell'acqua

8

8

Misura: una definizione precisa

- ◆ Misura è l'attribuzione di un numero (o di un simbolo) ad un attributo di una entità per caratterizzarlo
 - La misura non è un numero ma una relazione tra attributi e numeri !
- ◆ Per effettuare una misura dobbiamo avere una chiara idea di quali attributi vogliamo misurare e di quali entità possiedano tale attributo
 - La comprensione empirica precede la misura
 - Es. temperature, altezza, inflazione, ...
- ◆ È così nelle misure del software?

9

9

Misure dirette e indirette

- ◆ La misura diretta di un attributo non dipende da altri attributi
- ◆ Altrimenti parliamo di misure indirette
- ◆ Esempio:
 - Altezza di una persona: misura diretta
 - Temperatura di un corpo mediante termometro a mercurio: misura indiretta
 - » Misuriamo la lunghezza della barretta di mercurio

10

10

Sistema di relazioni empiriche

- ◆ Avere una comprensione empirica significa avere un insieme di relazioni (Re) che legano tra loro le entità (E) in esame
- ◆ Esempio:
 - Entità: personaggi dei fumetti
 - Attributo: altezza
 - Relazioni:
 - Re1(Pippo, Topolino) se “Pippo è più alto di Topolino”
 - Re2(Pippo) se “Pippo è alto”
 - Re3(Topolino, Pippo, Orazio) se “Topolino è più alto di Pippo quando siede sulle spalle di Orazio”

11

11

Sistema di relazioni numeriche

- ◆ Un sistema di relazioni numeriche è composto da un insieme di simboli (N) e da un insieme di relazioni (Rn) che legano tra loro tali simboli
- ◆ Esempio:
 - Insieme numerico: Numeri reali
 - Relazioni:
 - Rn1(x,y) se $x > y$
 - Rn2(x) se $x > 180$
 - Rn3(x,y,z) se $0.7x + 0.8z > y$

12

12

Misura

- ◆ Dato un sistema di relazioni empiriche e un sistema di relazioni numeriche si definisce “misura” una relazione M tra i due sistemi tale che:
 - ad ogni entità è associato un numero
 - ad ogni relazione tra entità corrisponde una relazione tra numeri
 - L' esistenza di una relazione tra istanze di entità implica l' esistenza della corrispondente relazione tra i numeri (misure) associate a quelle istanze di entità

13

13

Misura: esempio

- ◆ E = personaggio dei fumetti, R_e = relazioni di altezza, N = misure dell' altezza, R_n = relazioni tra numeri corrispondenti alle relazioni altezza.
 - Ad ogni personaggio è associato un numero (misura dell' altezza)
 - C' è un ordinamento tra numeri corrispondente alla relazione “ x è più alto di y ”, un ordinamento corrispondente a “ x è più alto di y quando siede sulle spalle di z ”, ecc.
 - Se Pippo (altezza 185) è più alto di Topolino (altezza 145), la coppia (185, 145) appartiene alla relazione numerica corrispondente ($x > y$).

14

14

Scala

- ◆ Dato un sistema di relazioni empiriche Σ , un sistema di relazioni numeriche Γ e una misura M si definisce “scala” la terna:

$$(\Sigma, \Gamma, M)$$

15

15

Esempio: criticità dei malfunzionamenti (1)

- ◆ Si supponga di dover misurare la criticità dei malfunzionamenti
 - Iniziamo con il classificare i malfunzionamenti in sintattici, semantici e crash del sistema. Supponiamo che ogni malfunzionamento appartenga ad una ed una sola di queste classi
 - Supponiamo che vi sia un ordinamento di criticità tale per cui i malfunzionamenti sintattici sono meno gravi di quelli semantici, che a loro volta sono meno gravi dei crash di sistema

16

16

Esempio: criticità dei malfunzionamenti (2)

- ◆ Sistema di relazioni empiriche:
 - E: insieme di tutti i possibili malfunzionamenti
 - $Re1(e)$ sse e è un malfunzionamento sintattico
 - $Re2(e)$ sse e è un malfunzionamento semantico
 - $Re3(e)$ sse e è un crash di sistema
 - $Re4(x,y)$ sse $(Re1(x) \wedge (Re2(y) \vee Re3(y))) \vee (Re2(x) \wedge Re3(y))$
- ◆ Sistema di relazioni numeriche:
 - $M(\text{malfunzionamento sintattico}) = 1$
 - $M(\text{malfunzionamento semantico}) = 2$
 - $M(\text{crash di sistema}) = 3$
 - $Rn4 \equiv <$

17

17

Trasformazioni di scala ammissibili

- ◆ L' esempio precedente ammette infiniti sistemi di relazioni numeriche e di conseguenza infinite misure e scale
 - $M(\text{malfunzionamento sintattico}) = 61$
 - $M(\text{malfunzionamento semantico}) = 106$
 - $M(\text{crash di sistema}) = 22342$
- ◆ Definiamo trasformazione di scala ammissibile una funzione che trasforma una qualsiasi scala in un'altra
- ◆ Il "tipo" di una scala è determinato dall' insieme di trasformazioni ammissibili

18

18

Principali tipi di scale

Trasformazioni possibili	Tipo scala	Esempi
$M' = F(M)$ F trasf. biettiva	Nominale	Label/classificazioni
$M' = F(M)$ F trasf. monotona	Ordinale	Preferenza, test di intelligenza
$M' = aM + b$ $a > 0$	Intervallo	Temperature
$M' = aM$ $a > 0$	Rateo	Durate, lunghezze
$M' = M$	Assoluta	Entità numerabili

19

19

Scala nominale

- ◆ Ogni attributo classificabile mediante un insieme di possibili valori dà luogo ad una scala nominale
- ◆ Esempi:
 - origine di malfunzionamenti
 - » gestione della memoria
 - » indicizzazione di vettori
 - » cattiva o mancata inizializzazione
 - linguaggio
 - » C
 - » Ada
 - » ...
- ◆ Non c'è relazione d'ordine

20

20

Scala ordinale

- ◆ Ogni attributo caratterizzato da un ordinamento lineare dei valori dà luogo ad una scala ordinale
- ◆ Esempio: criticità dei malfunzionamenti
 - vitale
 - grave
 - lieve
 - “opzione”
- ◆ Non sappiamo dire *quanto* “vitale” sia peggio di “grave”

21

21

Scala intervallo

- ◆ Se esiste anche un concetto di “distanza relativa tra i valori” abbiamo una scala intervallo
- ◆ Esempio:
 - indice di affidabilità del software
 - posso scegliere un valore “zero” di riferimento e dire che valori positivi indicano prodotti sempre più affidabili e valori negativi prodotti sotto la soglia

22

22

Scala rateo (rapporto)

- ◆ Se esiste anche un elemento “zero” siamo in presenza di attributi caratterizzati da scale rateo
- ◆ Esempi:
 - distanze,
 - intervalli di tempo,
 - la forza lavoro necessaria a sviluppare un pezzo di software,
 - ...

23

23

Scala assoluta

- ◆ Le scale assolute derivano da attributi che danno luogo ad un semplice conteggio di entità
- ◆ Esempio:
 - il numero di file che compongono un sistema
 - il numero è quello che è, e non ha senso “trasformarlo”.

24

24

Sensatezza di una affermazione

- ◆ La formalizzazione data sino ad ora ci permette di definire formalmente la “sensatezza” di una affermazione relativa ad una misura
- ◆ *Una affermazione relativa ad una misura è sensata se e solo se la sua verità resta invariata a fronte di una trasformazione ammissibile di scala*

25

25

Esempio di sensatezza

- ◆ Si considerino le seguenti affermazioni:
 - Il numero di errori scoperti in un programma è 100
 - » sensata, il numero di errori è una misura assoluta che non richiede l' indicazione della scala
 - Il costo di correzione di ogni errore in un programma x è 100
 - » priva di significato, il costo è una misura avente scala rateo.
 - » Occorre specificare la scala adottata
 - Un errore di allocazione di memoria richiede, per essere corretto, il triplo del tempo rispetto ad un errore di inizializzazione
 - » sensata, il tempo è una scala rateo ed è corretto parlare di “doppio” anche senza specificare la scala adottata

26

26

Tipi di scala ed operazioni ammesse

- ◆ Il tipo di scala di una misura determina l'insieme di operazioni ammesse
- ◆ Scale nominali e ordinali ammettono test statistici non parametrici:
 - percentile, mediana
- ◆ Scale intervallo e ratio ammettono test statistici parametrici:
 - Media, deviazione standard, media geometrica

27

27

Misure indirette trasformazioni di scala ammissibili

- ◆ Una misura indiretta di un attributo è una misura ottenuta misurando altri attributi
 - $M=f(M_1,...,M_n)$
- ◆ Una trasformazione di scala ammissibile per una misura indiretta M è una trasformazione T tale che:
 - $T(M)=f(T(M_1),...,T(M_n))$

28

28

La misura del software

- ◆ La principale lezione della teoria rappresentazionale della misura:
 - prima di misurare occorre decidere quali entità prendere in considerazione e quali attributi di tali entità misurare
- ◆ Come applicare tutto ciò al software:
 - Un modello concettuale (framework) per la misura del software

29

29

Software: cosa misurare

- ◆ Entità:
 - Processi
 - Prodotti
 - » Output dei processi
 - Risorse
 - » Input dei processi
- ◆ Attributi:
 - Attributi interni
 - Attributi esterni

30

30

Attributi interni ed esterni

◆ Attributi interni

- Sono gli attributi di una entità che possono essere misurati a partire dalla sola entità

◆ Attributi esterni:

- Sono gli attributi di una entità che devono essere misurati in termini di come l'entità è in relazione con l'ambiente esterno
- Sono i più importanti ma anche i più difficili da misurare

31

31

Esempi di entità ed attributi

ENTITÀ		ATTRIBUTI INTERNI	ATTRIBUTI ESTERNI
Prodotti	Specifica	Dimensione, riuso, modularizzazione, funzionalità, correttezza sintattica	Comprensibilità, manutenibilità
	Progetto	Dimensione, riuso, accoppiamento, coesione, modularizzazione, funzionalità	Qualità, comprensibilità, manutenibilità
	Codice	Dimensione, riuso, accoppiamento, modularizzazione, funzionalità, complessità algoritmica	Affidabilità, manutenibilità, usabilità
	Dati test	Dimensione, copertura	Qualità, affidabilità
Processi	Specifica	Durata, sforzo, n. cambiamenti nei requisiti	Qualità, costo, stabilità
	Progettazione	Durata, sforzo, numero di errori individuati	Costo, stabilità
	Test	Durata, sforzo, numero di errori trovati	Costo, controllabilità
Risorse	Personale	Età, costo	Produttività, esperienza
	Team	Dimensione, livello di comunicazione, strutturazione	Produttività
	Software	Costo, dimensione	Affidabilità
	Hardware	Costo, prestazioni	Affidabilità

32

32

Misure: valutazione vs. stima

- ◆ Si misura per valutare le caratteristiche di un oggetto esistente e misurabile
- ◆ Ma anche per stimare le caratteristiche di un oggetto non ancora esistente
- ◆ In entrambi i casi possiamo utilizzare un “modello” delle caratteristiche dell’oggetto che evidenzia relazioni esistenti tra diversi attributi dell’oggetto

33

33

Misure: valutazione vs. stima

- ◆ Esempio (COCOMO):
- ◆ $E = a S^b$
- ◆ Se E ed S sono noti il modello può essere utilizzato per valutare la corrispondenza del progetto alla media dei progetti dai quali derivano le costanti a e b
- ◆ Se S viene stimato, il modello può essere utilizzato per stimare E

34

34

Sistemi predittivi

- ◆ Un sistema predittivo è composto da un modello e da un insieme di procedure per predire i parametri incogniti del modello
- ◆ Esempio:
 - COCOMO è un sistema predittivo caratterizzato da un modello ($E = a S^b$) e da un insieme di procedure per predire i parametri a , b ed S

35

35

Sistemi predittivi vs. misure

- ◆ E' importante distinguere tra misure (usate a fini di valutazione) o sistemi predittivi
- ◆ Nel primo caso dovremo ricercare certe caratteristiche (rispetto della teoria rappresentazionale), nel secondo caso altre (bontà della stima effettuata)

36

36

Sistemi predittivi vs. misure: un esempio

- ◆ I punti funzione: misura delle funzionalità offerte da una applicazione o sistema predittivo dello sforzo di sviluppo?
 - Come misura delle funzionalità si rivelano molto scadenti
 - » Non corrispondono alla nozione empirica di “funzionalità” offerte da una applicazione
 - Come sistema predittivo (accompagnati da un modello FP/sforzo e da un insieme di procedure per stimare correttamente i punti funzione) presentano discrete caratteristiche

37

37

Comprendere la realtà prima di misurarla

- ◆ Spesso errori di misura derivano dal non aver individuato esattamente l'oggetto della misura (prodotto, processo, risorsa) e le caratteristiche dell'attributo che si misura (interno o esterno)
 - Esempio: si consideri l'attributo “numero di malfunzionamenti individuati durante la fase di test”
 - Tale attributo viene talvolta considerato un attributo di processo, tal altra un attributo di prodotto (numero di errori del programma)
 - » Si osservi che il primo è un attributo esterno (dipende dalla durata della fase di test), il secondo un attributo interno
 - Comprendere che si tratta di un attributo esterno di processo ci aiuta a capire da cosa possa dipendere (durata della fase di test, esperienza dei tester) e a costruire un modello adeguato

38

38

Progettazione degli esperimenti e analisi dei risultati

- ◆ Un esperimento di misura non può essere improvvisato
 - Occorre individuare ed isolare ogni aspetto che possa influire sui risultati
 - Esistono diverse linee guida che ci consentono di progettare al meglio gli esperimenti
- ◆ I risultati vanno analizzati usando gli opportuni metodi statistici anche in relazione al tipo della scala utilizzata per la misura

39

39

Esperimenti di misura: nozioni di base

- ◆ Un esperimento di misura si realizza per valutare gli effetti di certe procedure su certi soggetti
- ◆ Distinguiamo tra variabili indipendenti e variabili dipendenti
 - Con “variabili indipendenti” indichiamo la misura delle azioni compiute sul soggetto
 - Con “variabili dipendenti” indichiamo la misura di quegli attributi del soggetto che si ritiene possano essere influenzati dalle azioni compiute

40

40

Variabili dipendenti e indipendenti: un esempio

- ◆ Si supponga di voler valutare se: la produttività dei lavoratori è influenzata dalla intensità delle luci
- ◆ In questo esempio:
 - La produttività dei lavoratori è la variabile dipendente
 - L'intensità delle luci è la variabile indipendente

41

41

Strategie di sperimentazione (1)

- ◆ Durante un esperimento si sottopongono soggetti diversi a “trattamenti” diversi per cercare di mettere in evidenza un rapporto causale tra variabili dipendenti e indipendenti
- ◆ Un aspetto fondamentale della progettazione dell' esperimento consiste nello scegliere quali soggetti sottoporre a quali trattamenti al fine di isolare possibili elementi di influenza esterni
- ◆ Esistono diverse strategie che suggeriscono come effettuare tale selezione

42

42

Strategie di sperimentazione (2)

- ◆ Assegnamento casuale:
 - I soggetti sono assegnati ai diversi gruppi (sottoposti a trattamenti diversi) in maniera casuale
- ◆ Assegnamento a gruppi:
 - I soggetti simili (che si prevede rispondano in maniera simile) sono prima raggruppati, quindi i diversi gruppi sono assegnati in maniera casuale ai diversi esperimenti
 - Durante la fase di analisi statistica, le differenze tra i vari gruppi vengono tenute in considerazione per isolare l' influenza delle sole variabili indipendenti

43

43

Strategie di sperimentazione (3)

- ◆ Strategie di accoppiamento:
 - Pre-accoppiamento:
 - » In ogni gruppo (sottoposto a trattamento diverso) si fa in modo che esista un soggetto che sia stato preliminarmente accoppiato ad un soggetto degli altri gruppi in funzione della sua "somiglianza" con tali soggetti
 - Auto-accoppiamento:
 - » Ogni soggetto viene sottoposto a tutti i trattamenti (è come se appartenesse a tutti i gruppi)
 - » Bisogna fare attenzione che ogni soggetto non riceva i diversi trattamenti nello stesso ordine per eliminare possibili influenze derivanti dall' ordinamento dei trattamenti

44

44

Strategie di accoppiamento: esempio

- ◆ Si supponga di voler valutare l' impatto del linguaggio di programmazione adottato sul tempo di sviluppo
 - Pre-accoppiamento:
 - » Si prendono progetti simili per dimensione, complessità, esperienza del team e li si sviluppa ognuno con un linguaggio diverso
 - Autoaccoppiamento:
 - » Si sviluppa ogni applicazione più volte, ogni volta con un linguaggio diverso e si confrontano i risultati
 - » Come detto, occorre evitare di adottare i diversi linguaggi sempre nella stessa sequenza (prima C poi C++ ecc.) per eliminare l' influenza dell' ordine tra i linguaggi

45

45

Analisi dei risultati

- ◆ Durante la fase di analisi dei risultati la statistica ci permette di evidenziare quali variazioni nelle variabili dipendenti misurate riflettano veri rapporti causali con le variabili indipendenti
- ◆ Di norma per isolare tali “errori” si effettuano diverse misure delle variabili dipendenti per gli stessi valori delle variabili indipendenti e si esaminano i risultati ipotizzando una distribuzione gaussiana dell' errore

46

46

Progettazione di esperimenti: un esempio (1)

- ◆ Si vuole valutare se sia più semplice comprendere un algoritmo usando diagrammi di flusso o pseudo codice
 - Si decide di adottare una strategia di autoaccoppiamento: ogni soggetto vede tutti gli algoritmi, sia quelli espressi con pseudo codice sia quelli espressi tramite diagrammi di flusso e si misura il tempo di comprensione
 - Ovviamente, non potendo ripresentare lo stesso algoritmo tanto sotto forma di pseudo codice quanto sotto forma di diagramma di flusso si decide di avere algoritmi simili per complessità espressi in maniera diversa
 - » Si adottano tre classi di algoritmi, semplici, medi, complessi

47

47

Progettazione di esperimenti: un esempio (2)

- ◆ Al termine dell'esperimento si effettua un'analisi statistica evidenziando il tempo medio di comprensione di ogni classe di algoritmi da parte dei soggetti, quando gli algoritmi sono espressi tramite pseudo codice e quando sono espressi tramite diagrammi di flusso
 - Si noti come sia corretto utilizzare la media visto che la misura adottata (intervallo di tempo) presenta una scala rateo

48

48

Validazione delle misure del software

- ◆ Per il software sono state proposte numerose misure per diversi attributi
 - Spesso esistono molte misure diverse per lo stesso attributo
- ◆ Occorre una metodologia che ci permetta di valutare la bontà di tali misure
- ◆ La metodologia proposta si basa sulla distinzione tra misure e sistemi predittivi
 - Una misura è valida se caratterizza accuratamente l'attributo che si propone di misurare, rispondendo ai requisiti della teoria rappresentazionale
 - Un sistema predittivo è valido se permette una stima accurata

49

49

Validazione di un sistema di un sistema predittivo (1)

- ◆ La validazione di un sistema predittivo è il processo attraverso il quale si stabilisce l'accuratezza della stima fornita dal sistema predittivo
- ◆ Si fonda su un confronto dei risultati del sistema predittivo con i valori effettivi misurati a posteriori
 - L'errore percentuale diventa la misura principale dell'accuratezza del sistema predittivo
 - Esempio: Il COCOMO presenta un errore percentuale medio del 30%

50

50

Validazione di un sistema di un sistema predittivo (2)

- ◆ Talvolta la validità di sistemi predittivi complessi non è migliore di quella di sistemi più semplici
 - Esempio: Il sistema predittivo che afferma che per ogni giorno dell'anno a Londra piove è corretto nel 67% dei casi. I complessi modelli meteorologici forniscono previsioni corrette nel 70% dei casi

51

51

Validazione di una misura

- ◆ La validazione di una misura è il processo attraverso il quale si stabilisce se la misura fornisca una appropriata rappresentazione numerica dell'attributo
- ◆ Si basa sulla verifica del rispetto delle condizioni di rappresentazione
 - Esempio: una valida misura dell'attributo "lunghezza" di un programma non deve contraddire alcuna nozione empirica riguardante la nozione di lunghezza
 - » Ad esempio se $L(m)$ è la lunghezza del programma m deve valere la formula:
 - $L(m1;m2)=L(m1)+L(m2)$
 - » Analogamente per ogni $m1, m2$ per cui sia noto empiricamente che $m1$ è più lungo di $m2$ deve essere:
 - $L(m1)>L(m2)$

52

52

Validazione delle misure: alcuni errori comuni

- ◆ Nel campo del software si afferma spesso che una misura per essere valida deve essere un accurato predittore di un qualche attributo di interesse generale
- ◆ Per dimostrare la validità di una misura si cerca allora una correlazione con un qualche attributo “utile”
 - Senza aver prima in mente:
 - » Cosa si misura
 - » Perché lo si misura
 - » Quale sia la caratterizzazione empirica di tale attributo ritenuto “utile”
 - » Quale sia una ragionevole misura di tale attributo (fondamentale se la si vuole correlare con la misura da validare)
 - Esempio: LOC e complessità

53

53

La misura del software: un caso di studio

- ◆ Il numero ciclomatico di McCabe misura il numero di cammini linearmente indipendenti di un programma
 - Ciò spesso coincide con il numero di decisioni
- ◆ Il numero ciclomatico di McCabe viene ritenuto (senza alcun fondamento empirico) una valida misura della complessità di un programma
 - Una analisi di numerosi progetti ha mostrato una accurata correlazione tra il numero di linee di codice e il numero di McCabe
- ◆ La spiegazione risultante è stata:
 - “la complessità cresce al crescere della dimensione”
- ◆ Più esattamente si sarebbe dovuto dire:
 - “il numero di decisioni cresce al crescere della dimensione”
- ◆ Ma tale risultato è molto meno affascinante!

54

54

Raccolta e analisi dei dati

- ◆ Il processo di raccolta dei dati è fondamentale per garantire risultati accurati
 - Prima di iniziare un'attività di raccolta di dati bisogna avere chiaro in mente cosa vogliamo misurare e come (soprattutto nel caso di misure indirette)
 - I dati grezzi raccolti devono essere raffinati e quindi analizzati per fornire le misure previste
- ◆ Occorre decidere:
 - Come raccogliere i dati
 - Quando raccogliere i dati
 - Come conservare i dati e come estrarli quando utile
 - Come analizzare i dati

55

55

Come raccogliere i dati

- ◆ I dati possono essere raccolti attraverso procedure automatiche o manuali
 - Le prime, da preferirsi, sono possibili solo per pochi tipi di dati
- ◆ In ogni caso preliminarmente occorre decidere:
 - Quali entità e quali attributi misurare
 - Come effettuare le misure indirette (modelli)
 - Progettare le form per la raccolta manuale o le procedure per la raccolta automatica
 - Progettare le procedure per l'analisi dei dati e la presentazione dei risultati

56

56

Quando raccogliere i dati

- ◆ Dipende fortemente dall'entità e dall'attributo che occorre misurare
- ◆ In generale durante l'intero ciclo di vita
- ◆ Per alcuni dati (es. numero di malfunzionamenti rilevati durante l'uso) la raccolta prosegue anche dopo il rilascio del prodotto

57

57

Come conservare i dati

- ◆ I dati vanno conservati in un DB
 - Solo un DB fornisce gli strumenti per effettuare estrazioni efficienti
 - Inoltre l'uso di un DB riduce i problemi di accesso/modifica concorrente dei dati
- ◆ Lo schema adottato per il DB è estremamente critico
 - Da questo dipende la successiva possibilità di effettuare analisi e classificazioni accurate
 - Ecco perché è fondamentale aver deciso fin dall'inizio come si intende analizzare i dati per ricavare le misure (dirette e indirette) desiderate

58

58

Come analizzare i dati

- ◆ Per l'analisi dei dati ci serviamo dei metodi statistici
- ◆ Nel seguito si analizzeranno le principali tecniche per descrivere la distribuzione dei valori misurati per un singolo attributo e per individuare eventuali relazioni tra due attributi

59

59

Analisi dei dati: considerazioni preliminari

- ◆ Le misure nel campo del software presentano alcune peculiarità che non vanno trascurate
 - Come detto in precedenza non tutte le analisi statistiche sono applicabili a tutte le misure
 - » Dipende dal tipo di scala adottato
 - » Molte scale nel campo del software sono di tipo ordinale e non consentono quindi analisi statistiche quali il calcolo della media
 - Inoltre, molte analisi statistiche sono applicabili solo a distribuzioni normali mentre molte misure nel campo del software non hanno distribuzione normale
 - » Si può ricorrere ad analisi non parametriche o a statistiche "robuste"
 - » Si può cercare di variare la scala per avvicinarsi meglio a distribuzioni normali (es. scale logaritmiche invece che lineari)

60

60

Statistiche

- ◆ Vogliamo analizzare la distribuzione (supposta normale) delle misure di un singolo attributo:

- media:

$$\frac{\sum_{i=1..n} x_i}{n}$$

- varianza:

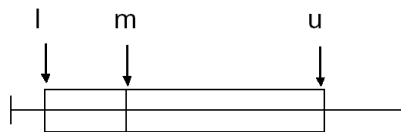
$$\frac{\sum_{i=1..n} (x_i - \bar{x})^2}{n - 1}$$

61

61

Statistiche: il box plot

- ◆ Distribuzioni non normali (o scale non rateo): statistiche robuste, il box plot
 - Mediana (m): valore del dataset che lo divide in due insiemi equipotenti (a destra della mediana c'è lo stesso numero di punti che sulla sinistra)
 - Upper fourth (u) e lower fourth (l): mediane dei valori maggiori di m e inferiori di m, rispettivamente
 - u, m e l dividono in 4 parti uguali il dataset.

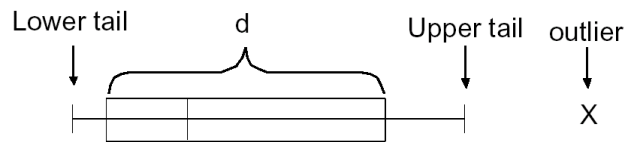


62

62

Il box plot

- ◆ Box length (d) = u-l
- ◆ Upper tail teorico: $u+1.5d$. È il valore che si dovrebbe trovare a quella distanza.
- ◆ Lower tail teorico: $l-1.5d$. È il valore che si dovrebbe trovare a quella distanza.
- ◆ I valori sono arrotondati all' elemento del dataset più vicino
- ◆ Le tail possono essere troncate



63

63

Box plot: esempio

ord	val
1	29
2	29
3	32
4	33
5	37
6	41
7	55
8	64
9	69
10	101
11	120
12	164
13	205
14	232
15	236
16	270
17	549

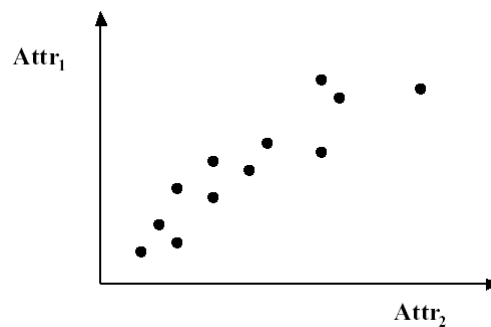
$$\begin{aligned}
 m_{\text{ord}} &= 17/2 = 9 \Rightarrow m_{\text{val}} = 69 \\
 l_{\text{ord}} &= m_{\text{ord}}/2 = 9/2 = 5 \Rightarrow l_{\text{val}} = 37 \\
 u_{\text{ord}} &= m_{\text{ord}} * 3/2 = 27/2 = 13 \Rightarrow u_{\text{val}} = 205 \\
 d_{\text{val}} &= u_{\text{val}} - l_{\text{val}} = 205 - 37 = 168 \\
 ut_{\text{val}} &= u_{\text{val}} + 1.5d_{\text{val}} = 205 + 1.5 * 168 = 457 \\
 lt_{\text{val}} &= l_{\text{val}} - 1.5d_{\text{val}} = 37 - 1.5 * 168 = -215, \\
 &\text{arrotondato a 29}
 \end{aligned}$$

64

64

Statistiche

- ◆ Per analizzare la relazione tra due attributi possiamo utilizzare uno “scatterplot”

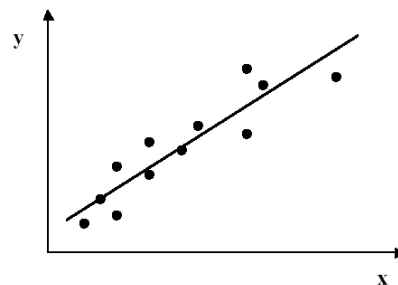


65

65

Statistiche per l'analisi delle relazioni tra due attributi

- ◆ Per valutare l'esistenza di una correlazione possiamo utilizzare la *regressione lineare*:
 - calcola la retta che meglio approssima l'insieme di valori
 - $y = a + b x$



- Le variabili indipendenti x possono essere molte.

66

66

Regressione lineare: esempio

- ◆ Supponiamo di dover ricavare una “regola” per stabilire il costo di un appartamento date alcune caratteristiche ritenute importanti:
 - Dimensioni (in m²)
 - DistanzaDalCentro (in km)
 - Età (in anni)
 - Classe (voto soggettivo da 1 a 10)
 - » si possono stabilire dei criteri basati su presenza di giardino, tennis, piscina, portineria, ecc.
- ◆ Supponiamo di avere un insieme di osservazioni e di volerne dedurre la regola.

67

67

Regressione lineare: esempio

Le osservazioni effettuate siano le seguenti:

App.	Dim.	Dist.	Età	Classe	Costo (ML)
1	120,0	0,5	10,0	5,0	999,0
2	60,0	1,2	12,0	6,0	745,0
3	75,0	1,3	8,0	7,0	885,0
4	150,0	4,6	20,0	6,0	992,0
5	40,0	3,0	16,0	7,0	621,0
6	55,0	2,5	6,0	6,0	782,0
7	110,0	0,8	2,0	6,0	1.092,0
8	132,0	2,6	18,0	6,0	938,0
9	100,0	4,5	22,0	5,0	736,0
10	80,0	3,0	7,0	7,0	854,0

68

68

Regressione lineare: esempio

Usiamo Excel per calcolare la correlazione lineare

	A	B	C	D	E	F
1	App.	Dim.	Dist.	Età	Classe	Costo (ML)
2	1	120,0	0,5	10,0	5,0	999,0
3	2	60,0	1,2	12,0	6,0	745,0
4	3	75,0	1,3	8,0	7,0	885,0
5	4	150,0	4,6	20,0	6,0	992,0
6	5	40,0	3,0	16,0	7,0	621,0
7	6	55,0	2,5	6,0	6,0	782,0
8	7	110,0	0,8	2,0	6,0	1.092,0
9	8	132,0	2,6	18,0	6,0	938,0
10	9	100,0	4,5	22,0	5,0	736,0
11	10	80,0	3,0	7,0	7,0	854,0
12						
13		17,382933	-11,7206	-13,0334	4,090605	554,3090648
14		10,698625	1,56264	6,838495	0,210581	77,48031039
15		0,9901059	19,3352	#N/D	#N/D	#N/D
16		125,08836	5	#N/D	#N/D	#N/D
17		187057,15	1869,25	#N/D	#N/D	#N/D

coefficienti della correlazione (blue line pointing to row 13)

coefficiente di determinazione (red line pointing to row 15)

69

Risultato della regressione

- ◆ $\text{Costo} = 554.30 + 4.09 \text{ Dimensioni} - 13.03 \text{ Distanza} - 11.72 \text{ Et\`a} + 17.38 \text{ Classe}$
- ◆ Ovviamente le dimensioni e la classe (che sono qualit\`a positive) hanno un coefficiente positivo, mentre distanza dal centro ed et\`a (qualit\`a negative) hanno coefficiente negativo.

70

70

Parametri calcolati da Excel

	A	B	C	D	E	F
1	m_n	m_{n-1}	...	m_2	m_1	b
2	se_n	se_{n-1}	...	se_2	se_1	se_b
3	r^2	se_y				
4	F	df				
5	ssreg	ssresid				

- ◆ se_1, se_2, \dots, se_n : i valori di errore standard per i coefficienti m_1, m_2, \dots, m_n .
- ◆ Se_b : l'errore standard per la costante b .
- ◆ r^2 : il coefficiente di determinazione.
- ◆ se_y : l'errore standard per la stima y .
- ◆ F: la statistica F, o il valore osservato F.
- ◆ df: il grado di libertà.
- ◆ ssreg: la somma di regressione dei quadrati.
- ◆ ssresid: la somma residua dei quadrati (SqErr)

71

71

Parametri calcolati da Excel (in Help della funzione REGR.LIN)

- ◆ se_1, se_2, \dots, se_n : i valori di errore standard per i coefficienti m_1, m_2, \dots, m_n .
- ◆ Se_b : il valore di errore standard per la costante b .
- ◆ r^2 : il coefficiente di determinazione. Confronta valori stimati e valori reali con un valore tra 0 e 1. Se è 1, c'è una correlazione perfetta nel campione: non c'è differenza tra il valore stimato e il valore reale. Se il coefficiente è 0, l'equazione di regressione non è utile per la stima.
- ◆ se_y : l'errore standard per la stima y .
- ◆ F: la statistica F, o il valore osservabile F. La statistica F è usata per determinare se la relazione osservata tra le variabili dipendenti e indipendenti è casuale.
- ◆ df: il grado di libertà, usato per trovare valori critici di F nella tabella statistica. Il confronto tra il valore trovato nella tabella e la statistica F prodotta dalla funzione REGR.LIN determina il livello di confidenza del modello.
- ◆ ssreg: la somma di regressione dei quadrati.
- ◆ ssresid: la somma residua dei quadrati.

72

72

Verifica della correlazione

	A	B	C	D	E	F	G	H	I	J
1	App.	Dim.	Dist.	Età	Classe	Costo (ML)	Stima	Errore	SqErr	Err Perc.
2	1	120,0	0,5	10,0	5,0	999,0	1008,4	9,4	87,9	0,9%
3	2	60,0	1,2	12,0	6,0	745,0	747,8	2,8	7,6	0,4%
4	3	75,0	1,3	8,0	7,0	885,0	872,1	-12,9	167,0	-1,5%
5	4	150,0	4,6	20,0	6,0	992,0	977,8	-14,2	200,7	-1,4%
6	5	40,0	3,0	16,0	7,0	621,0	613,0	-8,0	64,2	-1,3%
7	6	55,0	2,5	6,0	6,0	782,0	780,7	-1,3	1,7	-0,2%
8	7	110,0	0,8	2,0	6,0	1.092,0	1074,7	-17,3	299,1	-1,6%
9	8	132,0	2,6	18,0	6,0	938,0	953,7	15,7	246,8	1,6%
10	9	100,0	4,5	22,0	5,0	736,0	733,8	-2,2	4,9	-0,3%
11	10	80,0	3,0	7,0	7,0	854,0	882,1	28,1	789,3	3,2%
12										
13		17,382933	-11,7206	-13,0334	4,090605	554,3090648				
14		10,698625	1,56264	6,838495	0,210581	77,48031039				
15		0,9901059	19,3352	#N/D	#N/D	#N/D				
16		125,08836	5	#N/D	#N/D	#N/D				
17		187057,15	1869,25	#N/D	#N/D	#N/D				

73

73

Considerazioni sulla bontà della stima

- ◆ L' errore di stima varia tra -17,3 ML e +28,1 ML (da -1.6% a +3.1%)
- ◆ Nel 70% dei casi l' errore assoluto è inferiore (o uguale) al 1.5% (nel 90% dei casi è inferiore o uguale al 1.6%)
- ◆ L' errore medio si aggira intorno agli 11 ML
- ◆ NB: i criteri secondo cui valutare la bontà della stima dipendono da come si intende usare la stima stessa.

74

74

Considerazioni sull'affidabilità della stima

- ◆ Abbiamo ottenuto una stima accettabile. Ma è affidabile (o forse è buona “per caso”) ?
- ◆ $F = 125.08$, le variabili sono 4 (dimensioni, età, distanza, classe) e le osservazioni 10.
 - $v1 = k = 4$
 - $v2 = n - (k + 1) = 10 - (4 + 1) = 5$
- ◆ Con questi gradi di libertà $v1$ e $v2$, F deve essere > 5.19 (o > 11.39) per essere certi al 95 (99)% di probabilità che la stima non sia casuale.
- ◆ $F = 125.08 \gg 11.39$. Quindi siamo certi (per quanto si possa esserlo) che la nostra correlazione non sia dovuta a motivi casuali.

75

75

Considerazioni sulla correlazione

- ◆ I coefficienti sono tutti significativi?
 - Ad es. se avessi trascurato il dato sull'età dell'appartamento avrei ottenuto risultati diversi?
 - $t = m3/se3 = -11.72/1.56 = -7.51$
 - Consultando i valori tabulati della distribuzione t per $v2 = 5$ e per una confidenza del 90% si trova il valore 1.48. Poiché $7.51 > 1.48$ possiamo ritenere importante il contributo del coefficiente relativo all'età dello stabile.
 - Tutti gli altri parametri > 1.48 in valore assoluto: tutti i coefficienti sono rilevanti.
 - » $m1/se1 = 19.47$
 - » $m2/se2 = -1.90$
 - » $m4/se4 = 1.62$

76

76

Statistiche per l'analisi delle relazioni tra attributi

- ◆ Regressione logaritmica:
usata per calcolare la curva esponenziale che meglio approssima l'insieme di valori
- ◆ L'equazione della curva è:
 - $y = b * m^x$
 - $y = b * m_1^{x_1} * m_2^{x_2} * ...$ per molteplici variabili indipendenti
- ◆ Il modello usato da excel è
- ◆ $\ln y = x_1 \ln m_1 + ... + x_n \ln m_n + \ln b$
- ◆ Valgono gli stessi discorsi fatti per la regressione lineare.

77

77

Stabilire un programma di misura in ambito industriale

- ◆ Al fine di stabilire un programma di misura di successo in ambito industriale occorre tenere presenti diversi aspetti organizzativi:
 - Motivazioni ed obiettivi del programma
 - Responsabilità organizzative e composizione del team di misura
 - Attività di raccolta dei dati
 - Training e motivazione del personale

78

78

Motivazioni ed obiettivi del programma di misura (1)

- ◆ Ogni programma di misura deve avere dei chiari e precisi obiettivi
- ◆ Tali obiettivi dovrebbero essere espressi in termini di processi, prodotti e risorse ben precisi
 - Per facilitare la scelta delle metriche più adatte al raggiungimento degli obiettivi
- ◆ A partire dagli obiettivi dovrebbero essere definite anche le azioni da intraprendere a valle dell'analisi dei dati raccolti per raggiungere gli obiettivi
 - Un programma di misura che non abbia chiare le azioni da intraprendere per realizzare gli obiettivi non ha senso
- ◆ Ciclo di un programma di misura:
 - Misure → Fatti → Decisioni → Azioni

79

79

Motivazioni ed obiettivi del programma di misura (2)

- ◆ Esempio:
 - Obiettivo: migliorare la produttività del team di progetto
 - Soggetto: il team di progetto
 - Azioni: training, introduzione di nuove metodologie di progetto
- ◆ Nota finale:
 - Un programma di misura non ha possibilità di successo se manca il pieno supporto di tutto il personale coinvolto

80

80

Responsabilità e team di misura

- ◆ Per avere successo occorre avere chiaro chi abbia la responsabilità del progetto
- ◆ Per questo è opportuno creare un team di misura
 - Con responsabilità sul programma di misura
 - » Quali metriche da collezionare
 - » Quali tool utilizzare
 - » Come analizzare i dati raccolti
 - e l'opportuno potere che gli consenta di portare avanti il programma
- ◆ Il team di misura non deve però occuparsi di definire gli obiettivi del programma e le azioni da intraprendere (responsabilità del management)

81

81

Attività di raccolta dati

- ◆ A partire dagli obiettivi, una metodologia quale il GQM consente di ricavare le metriche da raccogliere
- ◆ Tali metriche devono essere adeguate agli obiettivi ma anche semplici da raccogliere per limitare l'impatto della raccolta dati sul processo di sviluppo
- ◆ Molte metriche sono di interesse generale
 - Dimensione dei prodotti e delle risorse, costi, sforzi
- ◆ e possono essere utilizzate in progetti diversi
- ◆ La loro raccolta deve essere standardizzata

82

82

Training e motivazione del personale

- ◆ Nessun programma di misura ha speranza di successo se il personale coinvolto non è opportunamente motivato
- ◆ Prima di iniziare il programma bisogna “venderlo”
 - Spiegare la necessità di misure
 - Pubblicizzare i vantaggi derivanti dalla raccolta dei dati
 - Spiegare come i dati verranno raccolti e come verranno utilizzati
- ◆ Successivamente occorre insegnare gli utenti come raccogliere i dati

83

83

Aspetti umani legati alla misura

- ◆ La maggior parte delle misure viene raccolta manualmente
 - La collaborazione del personale è fondamentale
- ◆ Alle persone non piace essere misurate
 - Specialmente se sulla base di tale misura esse sanno/pensano di essere valutate
- ◆ Devono essere chiari gli obiettivi del programma di misura
 - Nessuno verrà valutato sulla base dei dati raccolti
 - Il programma di misura porta dei vantaggi a tutta l'azienda
- ◆ Se possibile le misure devono essere raccolte in maniera anonima
 - Ad es. metriche per team invece che per singolo sviluppatore
- ◆ Alle persone non piace lavorare più del dovuto
 - Chiarire che opportune risorse aggiuntive verranno impiegate per minimizzare lo sforzo dell'attività di raccolta dati

84

84

Bibliografia

- ◆ Norman E. Fenton, Shari Lawrence Pfleeger, "Software Metrics: A Rigorous and Practical Approach", International Thomson Publishers, 3rd edition, Nov 2014.