Bilancia a due piatti: rapporto di calibrazione

Tommaso Raffaelli

‘23-12-2024

Contents

1 Overview dell’esperimento **2**

2 Preparazione dei dati **2**

3 Lettura dei dati **3**

4 Regressione **4**

4.1 Selezione dati di train 4

4.2 Regressione al minimi quadrati 5

4.3 Verifica di normalità 6

4.4 Analisi dei residui 6

4.5 Bootstrap 7

5 Conclusioni **8**

6 Tabelle **8**

List of Figures

* Tutte le tabelle troppo grandi per la visualizzazione all’interno del documento verranno messe per intero alla fine del documento

Chiave di verifica:

877b1af3fb86aea58f44c21692279e76cc770098

- Preparazione delle misure
- Preparazione dei dati
- Sviluppo del modello
- Controllo dei residui
- Comparazione con punti di test
- Test con bootstrap
- Conclusione
- Variabili modificabili

Durante la fase di test possiamo modificare due parametro della bilancia che sono:

- "Forza sul piatto sinisto"'
- `Differenza della forza fra i due piatti`

Il primo varia fra `100` e `500` a intervalli di `50` invece il secondo varia fra `-50` e`50` a intervalli di `5`. Generando una tabella di esecuzione delle misurazioni usando la funzione `expand.grid` si nota che essa è di dimensione `192 x ...`; essendo che il nostro esperimento è svolto digitalmente possiamo andare a prelevare tutte le combinazioni senza avere particolari conseguenze in termini di tempo e di spesa, cosa che in un test reale sarebbe da riconsiderare.

1 Overview dell'esperimento

In cosa consiste l'esperimento

Quale è l'obbiettivo dell'esperimento

2 Preparazione dei dati

Come detto in precendenza durante la taratura possiamo modificare due variabili del sistema *forza sul piatto sinistro* che chiameremo `Forza_sx` e *differenza fra i due piatti* che verrà chiamata `Delta_forza`. Deciso questo possiamo generare una tabella di test che verrà data all'operatore che svolgerà le misurazioni necessarie per la calibrazione

```
Ord_dati <- expand.grid(
  Forza_sx = seq(100, 500, 50),
  Delta_forza = seq(-50, 50, 5),
  Risultato = NA
) %>% mutate(
  StdOrd = 1:n(),
  RunOrd = sample(n()),
  .before = Forza_sx
) %>% arrange(RunOrd)

# write.csv(Ord_dati, "Test_suite.csv")
```

Definiamo `StdOrd` come ordine in cui `expand.grid` ha generato la tabella e `RunOrd` come una *casualizzazione* di `StdOrd` in modo da ottenere una sequenza di misure casulizzate così da diminurire l'effetto di di possibili variabili modificanti che non vendono considerate nel modello fisico.

Ottenendo la seguente tabella: 2.1:

```
Ord_dati[1:5,] %>%
  kable(booktabs=T, caption="Ordine di test") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), position = "center")
```

Table 2.1. Ordine di test

StdOrd	RunOrd	Forza_sx	Delta_forza	Risultato
100	1	100	5	NA
148	2	250	30	NA
70	3	400	-15	NA
51	4	350	-25	NA
65	5	150	-15	NA

OOOOOO Ora l'addetto alle misurazioni per la calibrazione può seguire fare le misurazioni in ordine casualizzato

3 Lettura dei dati

Come è già stato detto durante l'esperimento abbiamo *angle* come risultato, *F1* e *DF* come varibili da noi modificabili.

Tabella dei dati ottenuti: 3.1

```
Misurazioni <- read.csv("misurazioni.csv", comment="#")

Misurazioni[1:10,] %>%
  kable(booktabs=T, caption="Ordine di test") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), position = "center")
```

Table 3.1. Ordine di test

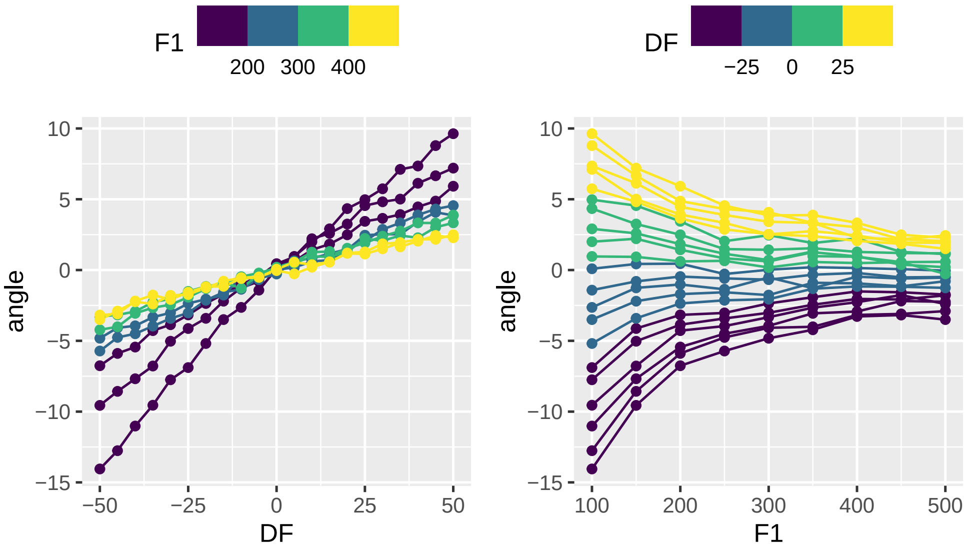
time	F1	DF	angle
9660	100	-50	-14.05
9720	100	-45	-12.76
9780	100	-40	-11.02
9840	100	-35	-9.55
9900	100	-30	-7.75
9960	100	-25	-6.89
10020	100	-20	-5.19
10080	100	-15	-3.50
10140	100	-10	-2.64
10200	100	-5	-1.42

Andato a mettere a grafico otteniamo:

```
g1 <- Misurazioni %>%
  ggplot(aes(x=DF, y=angle, group=F1, color=F1)) +
  geom_line() +
  geom_point(aes(y=angle)) +
  scale_color_viridis_b() +
  theme(legend.position="top")

g2 <- Misurazioni %>%
  ggplot(aes(x=F1, y=angle, group=DF, color=DF)) +
  geom_line() +
  geom_point(aes(y=angle)) +
  scale_color_viridis_b() +
  theme(legend.position="top")

g1 + g2
```



4 Regressione

Ora possiamo iniziare a fare la regressione però prima dobbiamo fare alcune considerazioni del tipo:

La regressione seguirà un modello fisico:

$$tan(angle) = (a\frac{DF}{2F1 + FM + DF})$$

questo non è un modello lineare infatti per poter regredire i punti dovremmo usare un metodo non lineare, nel nostro caso useremo il metodo dei minimi quadrati.

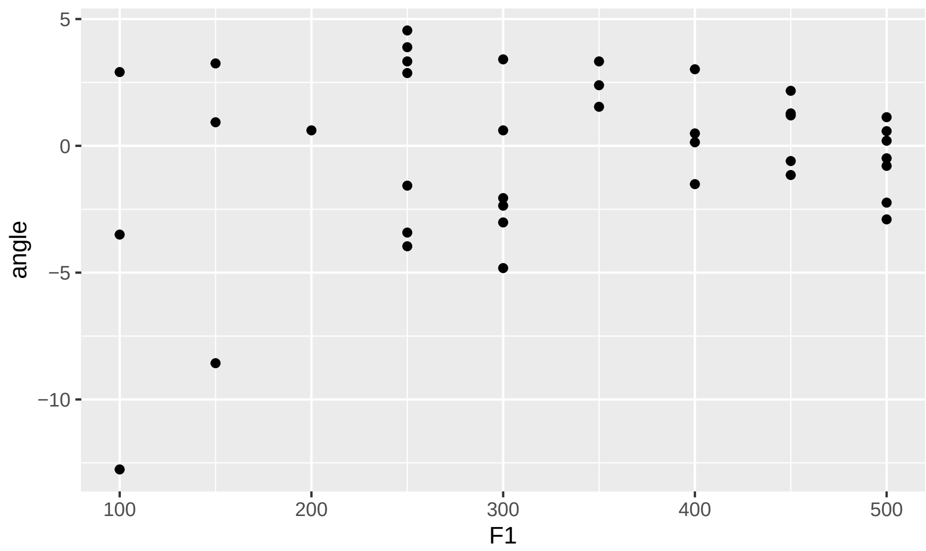
4.1 Selezione dati di train

Prima di fare la regressione possiamo selezionare alcuni punti da omettere dalla regressione del modello in modo da poterli usare come test.

```
Misurazioni <- Misurazioni %>%
  mutate(
    train = runif(n()) > 1/4
```

```
)

g1 <- Misurazioni %>%
  filter(!train) %>%
  ggplot(aes(x=F1, y=angle)) +
  geom_point()
g1
```




```
#g1 + wrap_table(Misurazioni[1:10,], space="fixed")
```

4.2 Regressione al minimi quadrati

Come già detto la regressione non sarà lineare ma useremo il metodo dei minimi quadrati che in *R* si utilizza tramite la funzione `nls`.

La funzione `nls` deve avere come argomenti:

- Funzione -> Funzione che rappresenta il modello fisico che stimo considerando per la taratura
- Start -> valore di partenza per le variabili di calibrazione `a` e `FM`
- Data set -> misurazioni su cui voliamo regredire il modello fisico

per comodità possiamo fare una funzione che contiene il modello fisico e ci restituisce l'algolo dati i parametri

```
func <- function(F1, DF, a, FM){
  angle <- a*(DF / 2*F1 + FM + DF)
  return(angle)
}
```

```
reg <- nls(tan(angle) ~ func(F1, DF, a, FM), start=list(a=1, FM=20), data=Misurazioni)
summary(reg)
```

```
##
## Formula: tan(angle) ~ func(F1, DF, a, FM)
```

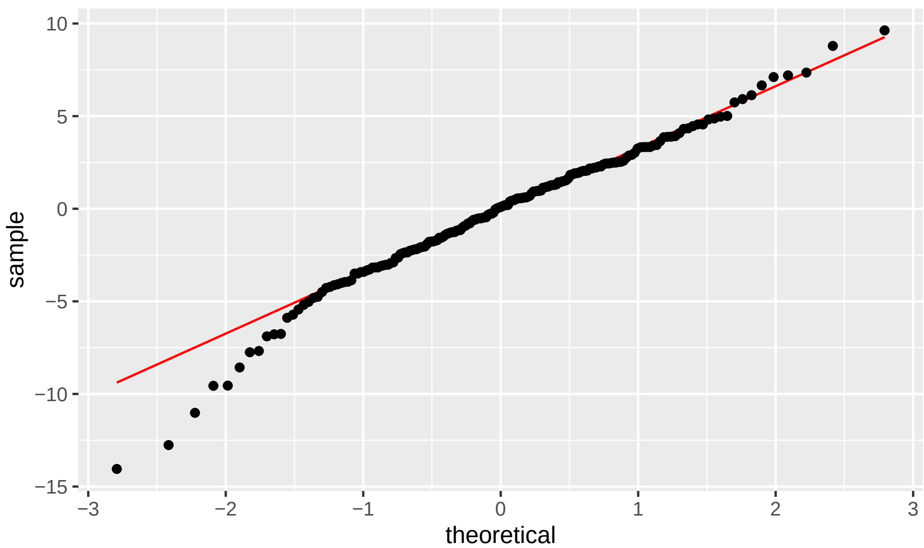
```
##
## Parameters:
##      Estimate Std. Error t value Pr(>|t|)
## a   2.000e-03  1.860e-03   1.075    0.284
## FM -6.408e+03  7.509e+03  -0.853    0.395
##
## Residual standard error: 128.3 on 189 degrees of freedom
##
## Number of iterations to convergence: 2
## Achieved convergence tolerance: 5.18e-08
```

```r
cor(atan(Misurazioni$angle), predict(reg)) %>% abs()
```

```
## [1] 0.8369042
```

4.3 Verifica di normalità

```r
Misurazioni %>%
  ggplot(aes(sample=angle)) +
  geom_qq_line(color="red") +
  geom_qq()
```



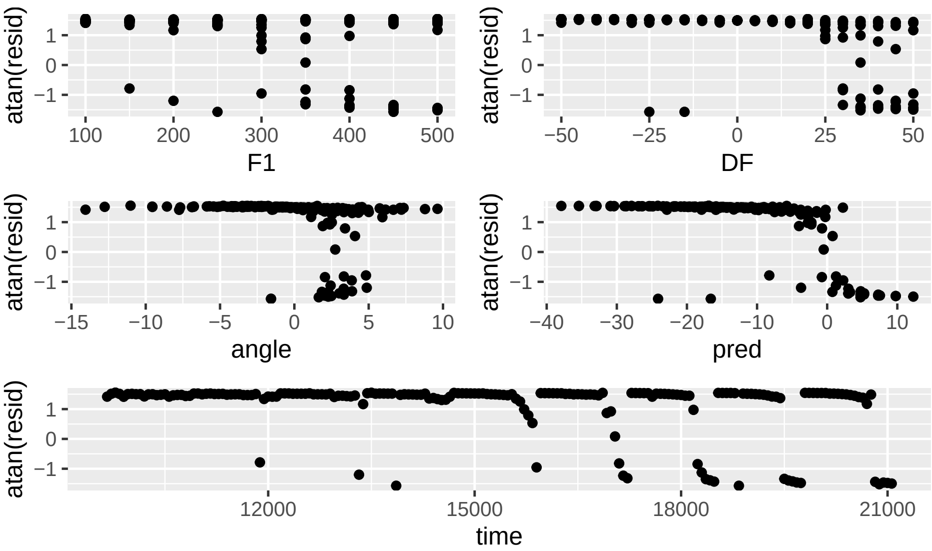

```r
shapiro.test(Misurazioni$angle)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Misurazioni$angle
## W = 0.97705, p-value = 0.003135
```

4.4 Analisi dei residui

```
resid_plot <- function(t, n) {
  t %>%
    ggplot(aes(x={{n}}, y=atan(resid))) +
    geom_point()
}


Misurazioni %>%
  add_residuals(reg) %>%
  add_predictions(reg) %>% {
    (resid_plot(., F1) + resid_plot(., DF)) /
    (resid_plot(., angle) + resid_plot(., pred)) +
     resid_plot(., time)
  }
```



4.5 Bootstrap

```
stats <- function(data){
  fit <- nls(tan(angle) ~ (a*(DF / 2*F1 + FM + DF)), data=data, start=list(a=1, FM=20))
  fit$m$getPars()
}

stats(Misurazioni)
```

```
##              a            FM
##   1.999870e-03 -6.407646e+03
```

```
Misurazioni.b <- boot(Misurazioni, R=10000, statistic=function(x, i) stats(x[i,]))
```

```
boot.ci(Misurazioni.b, type="perc", index=1)
```

```
## BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
## Based on 10000 bootstrap replicates
##
```

```
## CALL :
## boot.ci(boot.out = Misurazioni.b, type = "perc", index = 1)
##
## Intervals :
## Level     Percentile
## 95%   (-0.0001,  0.0058 )
## Calculations and Intervals on Original Scale
```

```
ci <- list(
  a = boot.ci(Misurazioni.b, type="perc", index=1)$percent[4:5],
  FM = boot.ci(Misurazioni.b, type="perc", index=2)$percent[4:5]
)
```

```
Misurazioni %>%
  mutate(
    #upper = map_dbl(t, f_conf(., f, ci)),
    #lower = map_dbl(t, f_conf(., f, ci, upper=F))
  ) %>% head()
```

```
##   time  F1  DF  angle train
## 1 9660 100 -50 -14.05  TRUE
## 2 9720 100 -45 -12.76 FALSE
## 3 9780 100 -40 -11.02  TRUE
## 4 9840 100 -35  -9.55  TRUE
## 5 9900 100 -30  -7.75  TRUE
## 6 9960 100 -25  -6.89  TRUE
```

5 Conclusioni

6 Tabelle

```
Misurazioni %>%
  head() %>%
  kable(booktabs=T, caption="Ordine di test") %>%
  kable_styling(latex_options = c("striped", "HOLD_position"), position = "center")
```

Table 6.1. Ordine di test

time	F1	DF	angle	train
9660	100	-50	-14.05	TRUE
9720	100	-45	-12.76	FALSE
9780	100	-40	-11.02	TRUE
9840	100	-35	-9.55	TRUE
9900	100	-30	-7.75	TRUE
9960	100	-25	-6.89	TRUE

877b1af3fb86aea58f44c21692279e76cc770098