Università di Pisa

SCUOLA DI INGEGNERIA

Corso di Laurea in Artificial Intelligence and Data Engineering

Statistica

Candidato
Marco Parola

Relatore
Prof. Romito Marco

Anno Accademico 2019–2020

Indice

1 Introduzione

Il dataset utilizzato per l'analisi dei dati del primo modulo, scaricabile dalla piattaforma kaggle al seguente link https://www.kaggle.com/harlfoxem/datasets , descrive vendite di case della Contea di King (King County) nel periodo compreso tra maggio 2014 e maggio 2015.

Numerosità della tabella: 21613
Numerosità dei fattori: 21

- Id, numero univoco per ogni vendita
- Date, data della vendita
- Price, prezzo della vendite
- Bedrooms, numero delle camere
- Bathrooms, numero di bagni presenti nella case, dove 0.5 indica la presenza del bagno, ma non della doccia.
- Sqft_living, metri quadri calpestabili.
- Sqft_lot, metri quadri totali
- Floors, numero di piani
- Waterfront, valore booleano, che indica se la casa è posizionata lungomare
- View, indice compreso tra 0 e 4 che indica la visuale
- Condition, indice compreso tra 1 e 5 che indica la condizione della casa
- Grade, indice compreso tra 1 e 13, dove valori 1-3 indicano una scarsa progettazione e costruzione dell'edificio, 7 è il valore medio, 11-13 indicano una ottimo progettazione e costruzione
- Sqft_above, metri quadri sopra il piano terra
- Sqft_basement, metri quandri al piano terra
- Yr_built, anno della costruzione
- Yr_renovated, anno dell'utima ristrutturazione
- Zipcode, codice di avviamento postale (CAP)
- Lat, latidune
- Long, Longitudine
- Sqft_living15, media dei metri quadri calpestabili delle 15 case più vicine
- Sqft_lot15, media dei metri quadri totali delle 15 case più vicine

L'obiettivo di questa analisi è poter fornire a una compagnia immobiliare un modello per poter dare una valutazione delle abitazioni di una famiglia di ceto medio, in relazione alle vendite effettuate precedentemente, per poter proporre dei prezzi competitivi sul mercato.

2 Preprocessing

Ho deciso di non considerare i seguenti attributi, perchè di poco interesse per l'analisi, che vogliamo effettuare: Date, Lat, Long, Zipcode, Sqft_basement. Dunque dopo aver importato la tabella ho visualizzato la correlazione tra gli attributi, al fine di avere una panoramica.

```
house = read.csv("house.csv");
house$date <- NULL; house$lat <- NULL; house$long <- NULL;
house$zipcode <- NULL; house$sqft_basement <- NULL;
round(cor(house), 2);
```

	id	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	sqft_living15	sqft_lot15
id	1	-0.02	0	0.01	-0.01	-0.13	0.02	0	0.01	-0.02	0.01	-0.01	-0.01	0.02	-0.02	0	-0.14
price	-0.02	1	0.12	0.23	0.32	0.03	0.1	0.13	0.22	0.02	0.3	0.27	0.16	0.01	0.06	0.26	0.04
bedrooms	0	0.12	1	0.52	0.58	0.03	0.18	-0.01	0.08	0.03	0.36	0.48	0.3	0.15	0.02	0.39	0.03
bathrooms	0.01	0.23	0.52	1	0.75	0.09	0.5	0.06	0.19	-0.12	0.66	0.69	0.28	0.51	0.05	0.57	0.09
sqft_living	-0.01	0.32	0.58	0.75	1	0.17	0.35	0.1	0.28	-0.06	0.76	0.88	0.44	0.32	0.06	0.76	0.18
sqft_lot	-0.13	0.03	0.03	0.09	0.17	1	-0.01	0.02	0.07	-0.01	0.11	0.18	0.02	0.05	0.01	0.14	0.72
floors	0.02	0.1	0.18	0.5	0.35	-0.01	1	0.02	0.03	-0.26	0.46	0.52	-0.25	0.49	0.01	0.28	-0.01
waterfront	0	0.13	-0.01	0.06	0.1	0.02	0.02	1	0.4	0.02	0.08	0.07	0.08	-0.03	0.09	0.09	0.03
view	0.01	0.22	0.08	0.19	0.28	0.07	0.03	0.4	1	0.05	0.25	0.17	0.28	-0.05	0.1	0.28	0.07
condition	-0.02	0.02	0.03	-0.12	-0.06	-0.01	-0.26	0.02	0.05	1	-0.14	-0.16	0.17	-0.36	-0.06	-0.09	0
grade	0.01	0.3	0.36	0.66	0.76	0.11	0.46	0.08	0.25	-0.14	1	0.76	0.17	0.45	0.01	0.71	0.12
sqft_above	-0.01	0.27	0.48	0.69	0.88	0.18	0.52	0.07	0.17	-0.16	0.76	1	-0.05	0.42	0.02	0.73	0.19
sqft_basement	-0.01	0.16	0.3	0.28	0.44	0.02	-0.25	0.08	0.28	0.17	0.17	-0.05	1	-0.13	0.07	0.2	0.02
yr_built	0.02	0.01	0.15	0.51	0.32	0.05	0.49	-0.03	-0.05	-0.36	0.45	0.42	-0.13	1	-0.22	0.33	0.07
yr_renovated	-0.02	0.06	0.02	0.05	0.06	0.01	0.01	0.09	0.1	-0.06	0.01	0.02	0.07	-0.22	1	0	0.01
sqft_living15	0	0.26	0.39	0.57	0.76	0.14	0.28	0.09	0.28	-0.09	0.71	0.73	0.2	0.33	0	1	0.18
sqft_lot15	-0.14	0.04	0.03	0.09	0.18	0.72	-0.01	0.03	0.07	0	0.12	0.19	0.02	0.07	0.01	0.18	1

Figura 1: Correlazioni tra gli attributi della tabella

Da una prima analisi qualitativa del dataset sono emersi i seguenti problemi:

- una diversa unità di misura per esprimere il valore del prezzo, alcuni sono espressi in migliaia di dollari, altri in dollari
- per alcune osservazioni i metri quadri calpestabili sono maggiori dei metri quadri totali
- presenza di outlier, in particolare ho trovato un campione relativo ad una casa di 33 camere, che è più attribuibile a un hotel o una villa di lusso, dunque possiamo considerare tale campione come un outlier ai fini del nostro studio.

Il codice seguente mostra come sia possibile riportare i valori del prezzo nella stessa unità di misura:

```
for(i in 1:length(house[,1])){
    if(house$price[i] > 4000000){
        house$price[i] = house$price[i] / 1000;
    }
}
```

Il codice seguente mostra come sia possibile eliminare tutti i campioni che hanno valori inconsistenti sulla metratura e quelli che hanno un numero di camere maggiori di 30 (parametro fissato euristicamente, in cui si fa riferimento ad abitazioni non ordinarie). Dopo l'esecuzione della seguente porzione di codice, il dataset conterrà 20824 campioni.

```
a = c();

for(i in 1:length(house[,1])){
    if((house$sqft_living[i] > house$sqft_lot[i]) | house$bedrooms > 30){
        a = c(a, i);
    }
}

house = house[-a,];
```

3 Modello lineare

3.1 Calcolo del modello lineare

Il passo successivo ha previsto il calcolo del modello della regressione lineare, considerando tutti gli attributi come parametri di ingresso e la visualizzazione del risultato ottenuto.

```
house.lm = lm(price~.,data=house);
summary(house.lm);
```

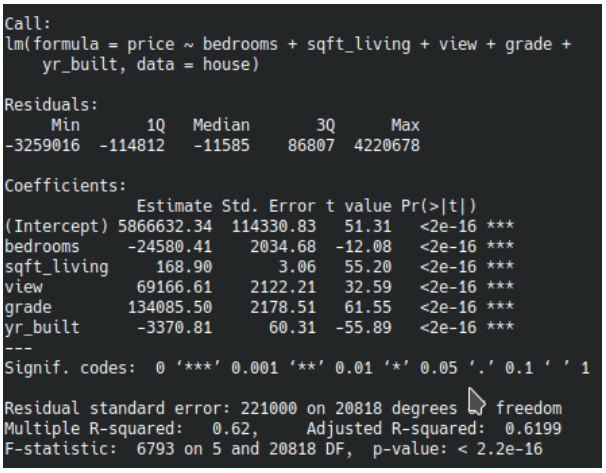
```
Call:
lm(formula = price ~ ., data = x)

Residuals:
     Min       1Q   Median       3Q      Max
-3159076  -109899   -12503    85722  4338312

Coefficients: (1 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    6.249e+06  1.403e+05  44.552  < 2e-16 ***
id            -2.585e-06  5.264e-07  -4.911 9.13e-07 ***
bedrooms      -2.964e+04  2.042e+03 -14.518  < 2e-16 ***
bathrooms      4.521e+04  3.575e+03  12.644  < 2e-16 ***
sqft_living    1.306e+02  4.701e+00  27.776  < 2e-16 ***
sqft_lot       1.104e-02  5.112e-02   0.216 0.828964
floors         1.471e+04  4.182e+03   3.517 0.000437 ***
waterfront     4.669e+05  1.873e+04  24.927  < 2e-16 ***
view           4.471e+04  2.292e+03  19.508  < 2e-16 ***
condition      2.088e+04  2.491e+03   8.383  < 2e-16 ***
grade          1.216e+05  2.292e+03  53.052  < 2e-16 ***
sqft_above     2.857e+00  4.718e+00   0.606 0.544809
sqft_basement         NA         NA      NA       NA
yr_built      -3.602e+03  7.185e+01 -50.128  < 2e-16 ***
yr_renovated   1.030e+01  3.916e+00   2.630 0.008539 **
sqft_living15  4.611e+01  3.623e+00  12.724  < 2e-16 ***
sqft_lot15    -5.072e-01  7.820e-02  -6.487 8.96e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 215100 on 20808 degrees of freedom
Multiple R-squared:   0.64,     Adjusted R-squared:  0.6397
F-statistic:  2466 on 15 and 20808 DF,  p-value: < 2.2e-16
```

Figura 2: Riepilogo modello regressione lineare con tutti i fattori

Dal risultato si nota che la varianza spiegata ammonta al 64%, inoltre considerando i p-value dei vari attributi si deduce che alcuni fattori sono ridondanti.
E' ragionevole ridurre i fattori di ingresso tenendo conto dei p-value e delle correlazioni presenti tra i vari attributi; in particolare eliminando i fattori uno ad uno, ho selezionato i seguenti attributi ed ho rieseguito la regressione:

- Bedrooms
- Sqft_living
- View
- Grade
- Yr_built

```
house.lm=lm(price~bedrooms+sqft_living+view+grade+yr_built,data=house);
summary(house.lm);
```



```
Call:
lm(formula = price ~ bedrooms + sqft_living + view + grade +
    yr_built, data = house)

Residuals:
     Min       1Q   Median       3Q      Max
-3259016  -114812   -11585    86807  4220678

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 5866632.34  114330.83   51.31   <2e-16 ***
bedrooms     -24580.41    2034.68  -12.08   <2e-16 ***
sqft_living     168.90       3.06   55.20   <2e-16 ***
view          69166.61    2122.21   32.59   <2e-16 ***
grade        134085.50    2178.51   61.55   <2e-16 ***
yr_built      -3370.81      60.31  -55.89   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 221000 on 20818 degrees of freedom
Multiple R-squared:   0.62,     Adjusted R-squared:  0.6199
F-statistic:  6793 on 5 and 20818 DF,  p-value: < 2.2e-16
```

Figura 3: Riepilogo modello regressione lineare ridotto a 5 fattori

Per tali parametri in input il modello ha perso il 2% della varianza spiegata, perdita accettabile, considerando che il modello è stato ottimizzato da 17 fattori a 5.
Ho effettuato anche la standardizzazione della tabella con il comando *scale()*, ma questo non ha variato il grafico di dispersione, per cui ho eseguito l'analisi utilizzando la tabella come era data.

4 Valutazione del modello lineare

Lo step successivo è la valutazione della bontà del modello, per fare ciò si calcolano i residui del modello e si analizzano, con l'obiettivo di non trovare una struttura nei dati, ma un comportameto "casuale". Questo significherebbe che siamo riusciti ad estrarre tutta l'informazione dei dati e nei residui è rimasta solo una componente che possiamo attribuire a rumore.

4.0.1 Plot dei residui del modello lineare

Il seguente codice mostra come calcolare i residui del modello precedentemente calcolato e visualizzarlo.

```
house.lm.r = residuals(house.lm);
plot(house.lm.r, pch=".");
```

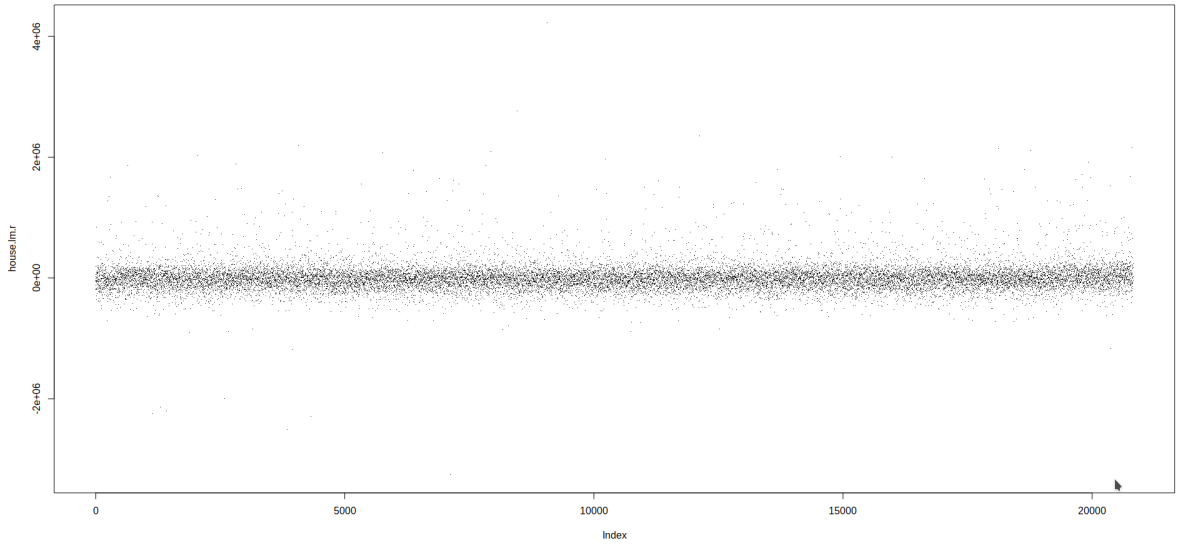


Figura 4: Grafico dei residui

Dal grafico dei residui si nota come i valori non posseggano una struttura, ma siano disposti come una nuvola di punti uniforme intorno allo zero, dunque nel complesso la media si può considerare nulla.
L'andamento random dei residui si può confermare se questi possiedono una distribuzioe gaussiana; in seguito approfondiremo la questione con due diversi approcci grafici. Non è stato svolto il test di Shapiro-Wilk, in quanto l'implementazione di tale test in R prende in un input un numero massimo di 5000 campioni e la nostra popolazione è molto più numerosa.

4.0.2 QQ-Plot modello lineare

Il primo test che effettuiamo per comprendere la natura della distribuzioe è il grafico qq-plot, un grafico in cui la distribuzione dei residui del nostro modello viene confrontato con una distribuzione normale. Se gli oggetti osservati hanno una distribuzione gaussiana, i punti si addenseranno lungo una retta.

```
qqnorm(house.lm.r, pch="*");
qqline(house.lm.r, pch=".", col="red");
```

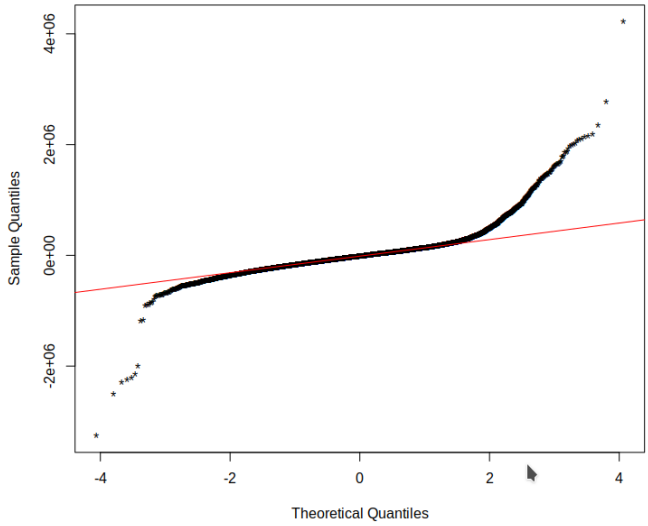


Figura 5: Grafico QQ-plot dei residui del modello lineare

Dal grafico in figure 5 si nota come i residui seguano un andamento lineare per buona parte dei valori, ma le code (in particolar modo quella superiore) si discostano dalla retta teorica che vorremmo che seguissero.
Dunque si può ipotizzare che forse il modello lineare non è il migliore per la nostra analisi, ma per trarre le conclusioni eseguiremo un secondo test.

4.0.3 Grafico della distribuzione dei residui

In figura 6 viene messo a confronto la distrubuzione dei residui, con una distribuzione normale che ha come valor medio e deviazione standard, quelli calcolati dai residui; dunque confrontiamo il distribuzione dei residui, con il valore teorico che vorremmo che avessero, nella speranza che le due non siano così discrepanti.

```
hist(house.lm.r,40,freq=F, ylim=c(0, 3e-06), xlim=c(-2e+06, +2e+06));
lines(density(house.lm.r),col="red");
lines(sort(house.lm.r),dnorm(sort(house.lm.r),mean(house.lm.r),sd(house.lm.r)));
```

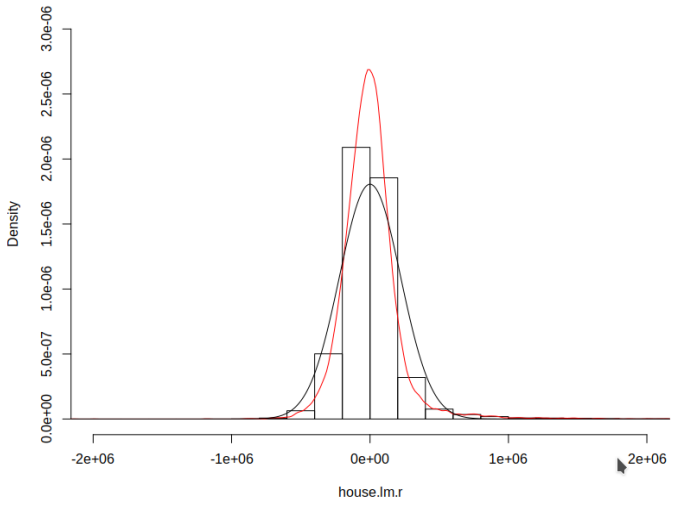


Figura 6: Grafico della distribuzione dei residui

Dal grafico si conferma l'ipotesi fatta nel precedente paragrafo, che con una regressione lineare si possono otteneare dati soddisfacenti solo in parte, per eseguire la previsione dei prezzi, infatti è evidente che i due andamenti presentino differenze.

5 Modello non lineare

E' opportuno calcolare un altro modello per migliorare i risultati, per testare se un modello non lineare è in grado di estrarre meglio informazioni dalla tabella, dunque introduremo i logaritmi.

5.1 Calcolo modello non lineare

```
house.lm_log = lm(log(price)~bedrooms+sqft_living+view+grade+yr_built ,data=house);
summary(house.lm_log);
```

```
Call:
lm(formula = log(price) ~ bedrooms + sqft_living + view + grade +
    yr_built, data = house)

Residuals:
    Min      1Q  Median      3Q     Max
-7.7047 -0.2165  0.0187  0.2212  1.2845

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.057e+01  1.791e-01 114.850   <2e-16 ***
bedrooms     2.048e-04  3.187e-03   0.064    0.949
sqft_living  1.860e-04  4.793e-06  38.818   <2e-16 ***
view         6.265e-02  3.324e-03  18.845   <2e-16 ***
grade        2.404e-01  3.412e-03  70.455   <2e-16 ***
yr_built    -4.958e-03  9.447e-05 -52.485   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3461 on 20818 degrees of freedom
Multiple R-squared:  0.5837,    Adjusted R-squared:  0.5836
F-statistic:  5837 on 5 and 20818 DF,  p-value: < 2.2e-16
```

Figura 7: Riepilogo modello di regressione non lineare

Il sommario del modello denota come la varianza spiegata sia diminuita al 58%, dunque in prima battuata un modello lineare parrebbe più adatto a descrivere il fenomeno, ma vale la pena effettuare i test che abbiamo eseguito anche per il modello lineare, per avere più fattori per valutare la bontà di questo secondo modello.

6 Valutazione modello non lineare

6.0.1 Plot dei residui del modello non lineare

Anche in questo caso come primo step calcoliamo e visualizziamo graficamente i residui del modello.

```
house.lm_log.r = residuals(house.lm_log);
plot(house.lm_log.r, pch=".", ylim=c(-3, 3));
```

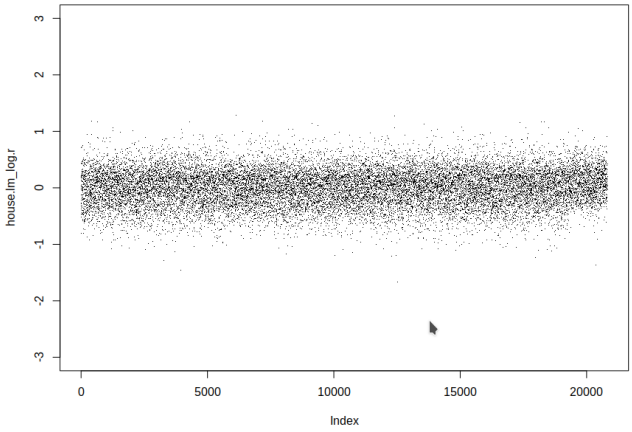


Figura 8: Grafico dei residui

Anche in questo caso si presenta una nuvola di punti uniforme, senza nessun tipo di struttura. La differenza che si nota con il grafico in figura 8 è il range tra cui variano i valori, nel secondo nettamente inferiore.

6.0.2 QQ-Plot modello non lineare

```
qqnorm(house.lm_log.r, pch="*", ylim=c(-7,2));
qqline(house.lm_log.r, pch=".", col="red");
```

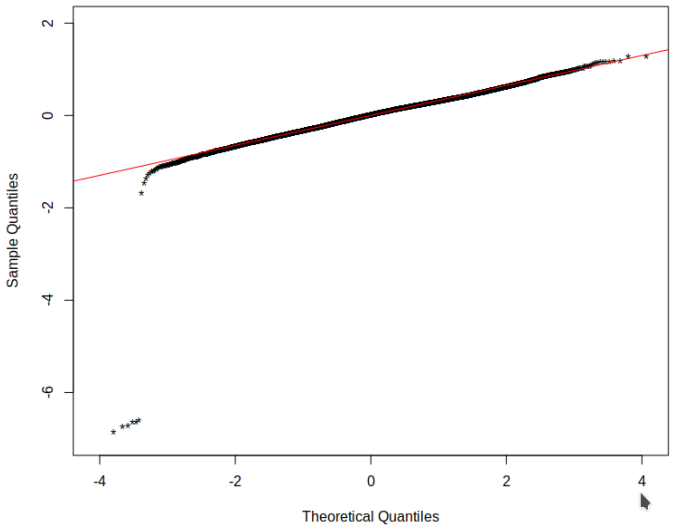


Figura 9: Grafico QQ-plot dei residui del modello non lineare

Il qq-plot di questi residui è molto soddisfacente, perchè fatta eccezione per alcuni outlier presenti nella parte inferiore del grafico, i punti seguono un andamento lineare.

6.0.3 Grafico della distribuzione dei residui

Similmente a quanto fatto per il modello lineare, anche in questo caso è necessario confrontare l'andamento della distribuzione teorico, con quello osservato.

```
hist(house.lm_log.r,40,freq=F, xlim=c(-2,2));
lines(density(house.lm_log.r),col="red");
lines(sort(house.lm_log.r),dnorm(sort(house.lm_log.r),mean(house.lm_log.r),sd(house.lm_log.r)));
```

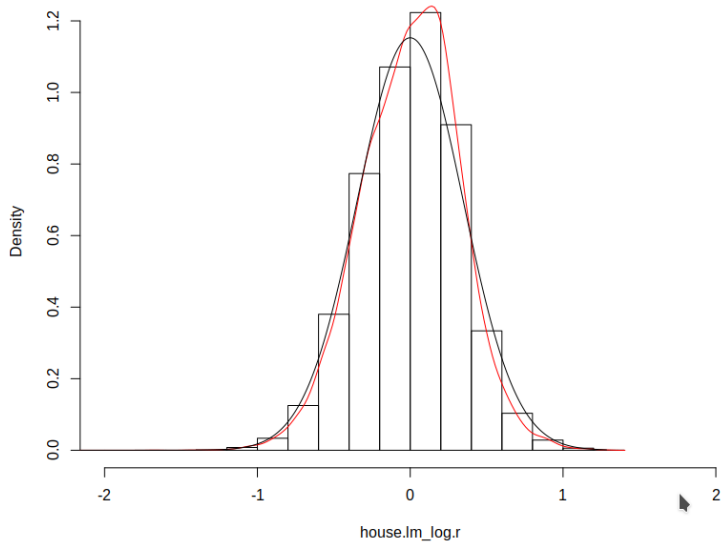


Figura 10: Grafico della distribuzione dei residui

E' evidente che in questo caso le due distribuzioni siano molto più affini, ma non si può ancora concludere che questo modello sia migliore del precedente, nonostante i residui abbiano una distribuzione gaussiana, poichè la varianza spiegata è inferiore, rispetto al primo modello.

7 Confronto tra i modelli

Lo step finale dell'analisi prevede il confronto dei due modelli, al fine di scegliere quello che è in grado di predire in maniera più accurata l'attributo prezzo. Avendo a disposizione un unico dataset, lo splittiamo in due parti trainig-set, contente l'80% delle osservazioni totali, e test-set, contente il 20%; dopo di che ricalcoliamo i modelli avvelendosi unicamente del training-set, prediciamo i valori del prezzo, dando in input al modello il test-set, infine confrontiamo i valori ottenuti e quelle predetti.

```
lhouse = log(house);
is.na(lhouse) <- sapply(lhouse, is.infinite)
lhouse[is.na(lhouse)] <- 0

set.seed(25)

u = sample( length(house[,1]), round(length(house[,1]) * 0.2));
houseTraining = house[-u,];
houseTest = house[u,];
house.lm = lm(price~bedrooms + sqft_living + view+grade + yr_built, data = houseTraining);
house.lm.p = predict(house.lm, houseTest);

lhouseTraining = lhouse[-u,];
lhouseTest = lhouse[u,]
lhouse.lm_log = lm(price~bedrooms + sqft_living + view+grade + yr_built, data = lhouseTraining);
lhouse.lm_log.p = predict(lhouse.lm_log, lhouseTest);

sqrt(mean((house.lm.p-houseTest$price)^2)/mean(houseTest$price)^2)
sqrt(mean((lhouse.lm_log.p-lhouseTest$price)^2)/mean(lhouseTest$price)^2)
```

Si è reso necessario il rimpiazzamento di alcuni valori del dataframe uguali a *-Inf*, in quanto abbiamo applicato il logaritmo a valori uguali a zero, questa operazione è svolta nelle prime tre righe del precedente codice.
Dal calcolo dell'errore relativo si scopre che il modello non lineare è quello che predice con molta più accuratezza, rispetto a quello lineare, i valori i prezzi in uscita; infatti predice con un errore solo del 2%.

```
> sqrt(mean((house.lm.p-houseTest$price)^2)/mean(houseTest$price)^2)
[1] 0.3960953
> sqrt(mean((lhouse.lm_log.p-lhouseTest$price)^2)/mean(lhouseTest$price)^2)
[1] 0.02565676
```

Figura 11: Errori relativi della predizione

8 Conclusioni

Si può concludere che il modello non lineare è uno strumento più adatto ad un'impresa immobiliare, rispetto a un modello lineare, per stimare il prezzo di una casa, avendo in input le sue caratteristiche. E' importante notare, però, che non è uno strumento universale, in quanto la varianza spiegata del modello logaritmico ammonta al 58%.
Dunque forniamo uno strumento di supporto decisionale, che dovrebbe essere utilizzato dalla compagnia per fare una prima stima del prezzo, dopo di che il valore predetto deve essere confermato o eventualmente aggiustato da chi lavora all'interno dell'impresa.