



# PREDITTORE DELLA DOMENICA

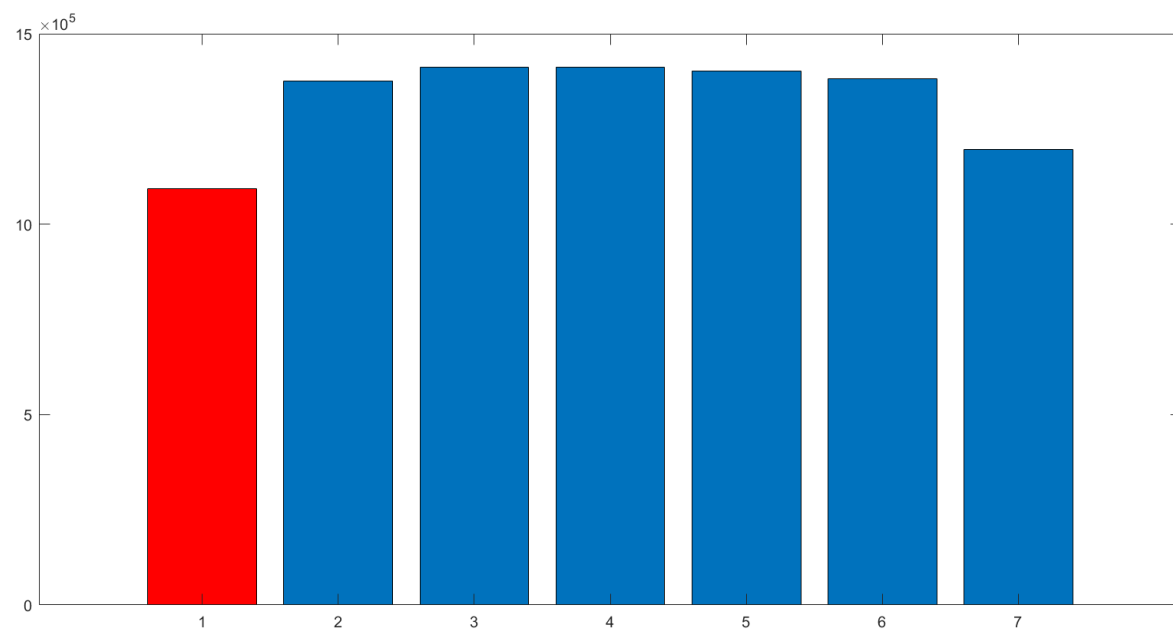
- Noemi Cardillo
- Antonio Coronelli
- Jacopo Del Col
- Federico Guareschi
- Simone Tartarotti



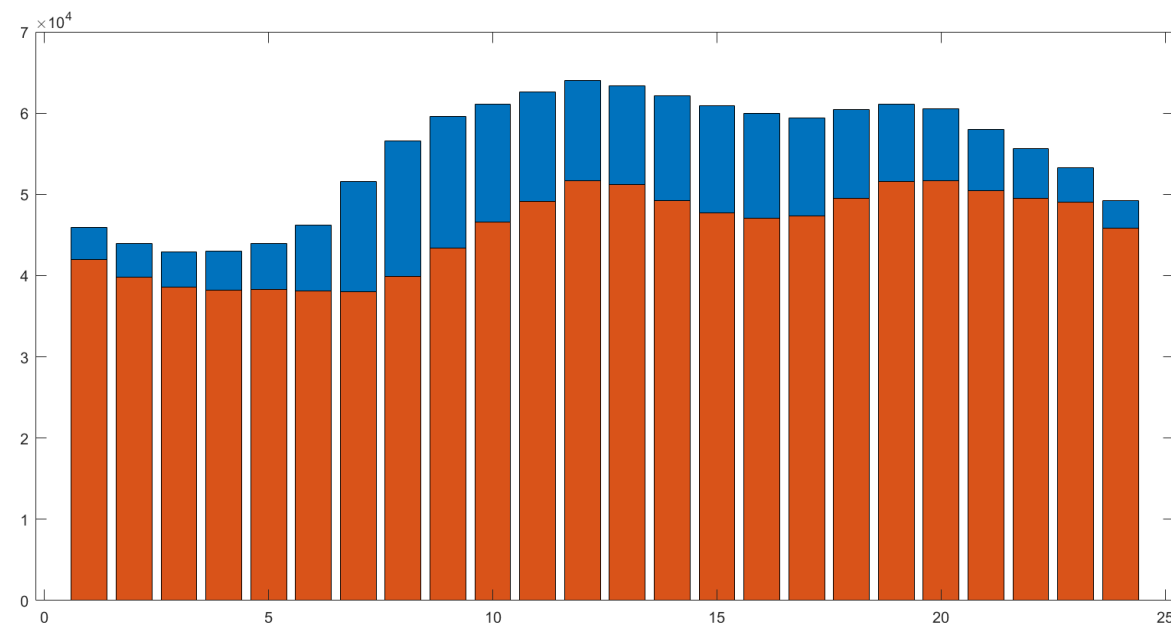
## Obiettivi

- Predire il consumo di ogni ora delle domeniche dell'anno successivo agli anni dati
- Realizzare una funzione che restituisca la previsione corrispondente all'ora e al giorno dell'anno (che deve essere una domenica) ricevuti in input

Nell'identificazione vengono considerati solo i consumi relativi alle domeniche. Si nota infatti che questi seguono un andamento diverso rispetto agli altri giorni:



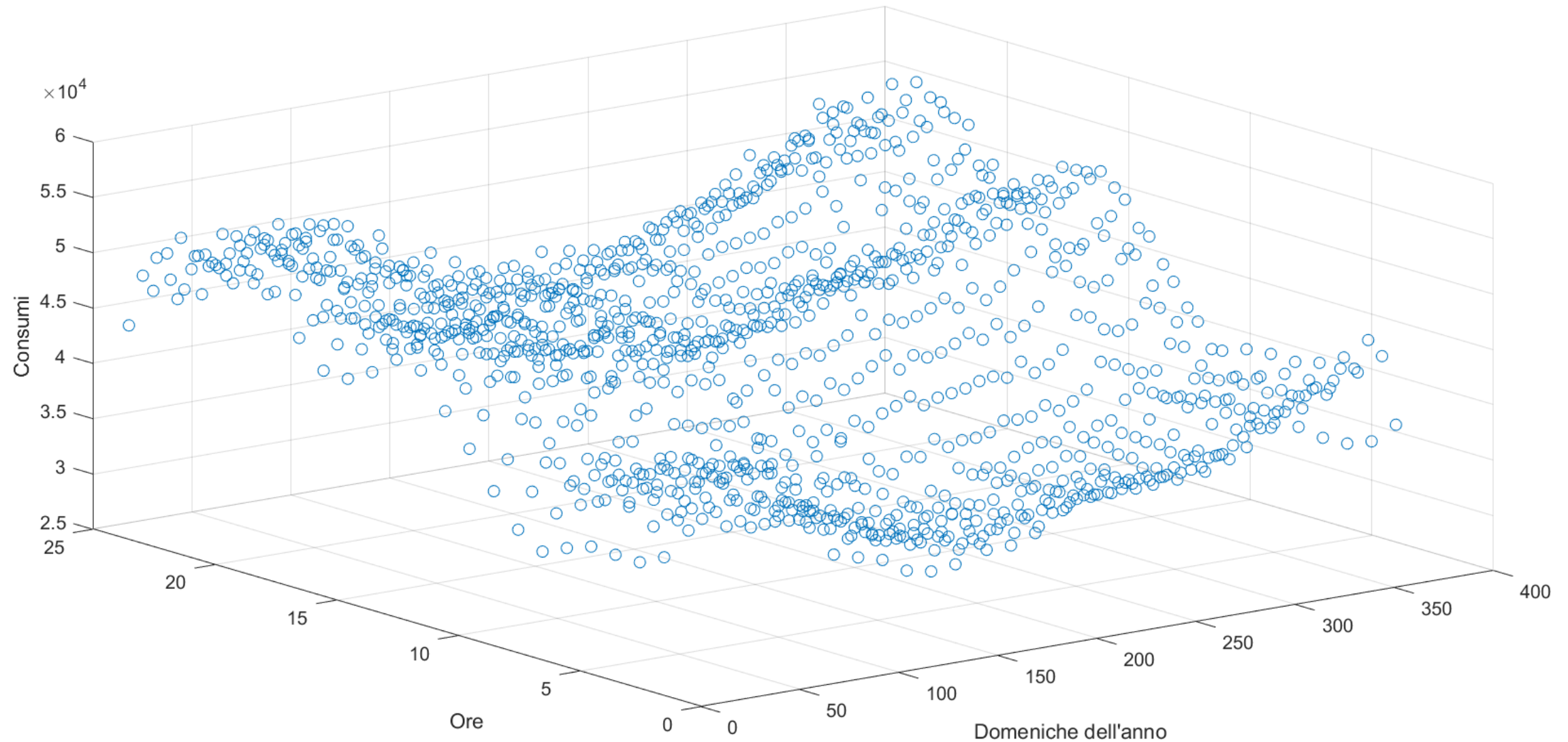
Media dei consumi di ogni giorno della settimana



Media dei consumi sulle 24 ore

*In rosso i consumi delle domeniche, in blu i consumi degli altri giorni.*

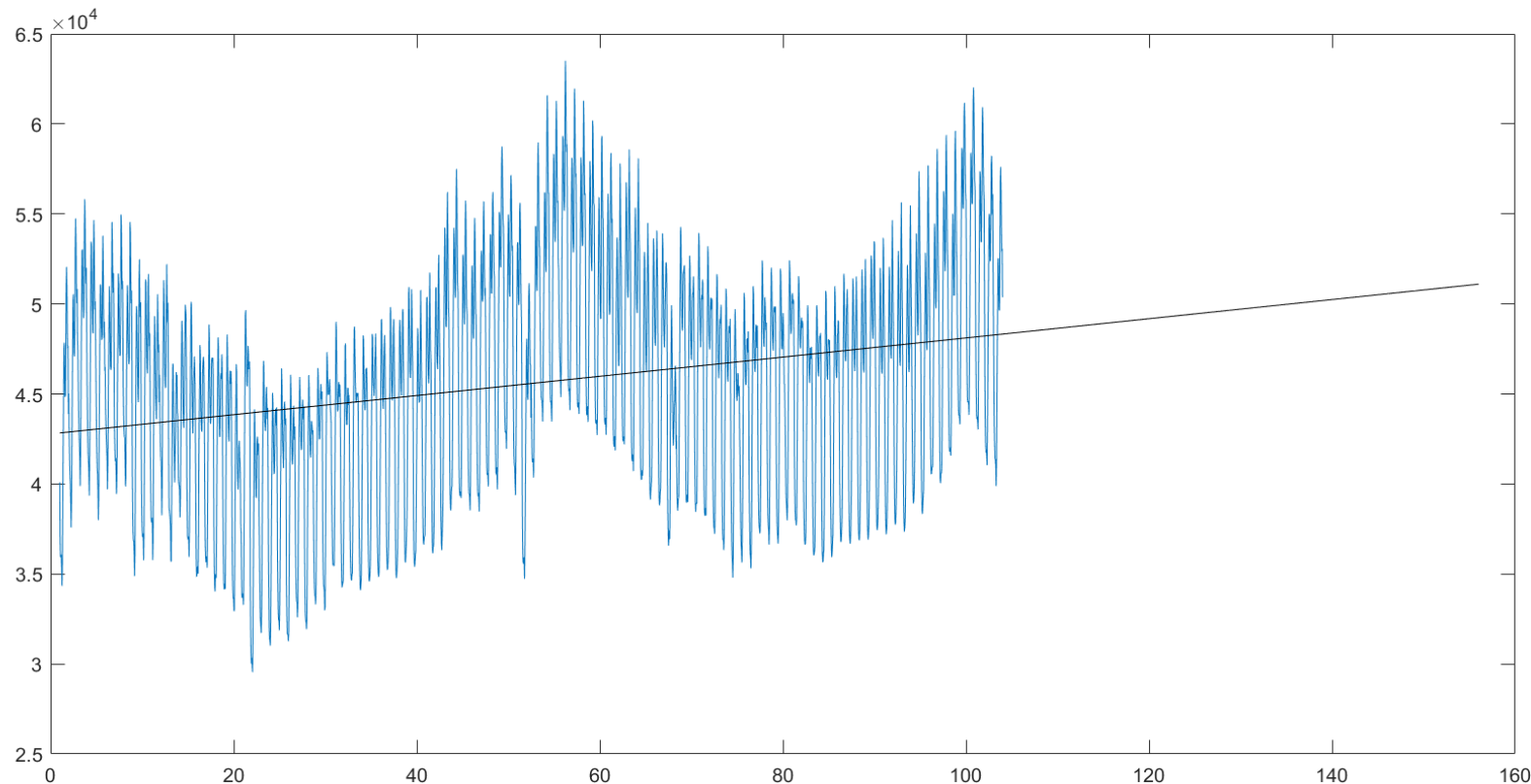
# CONSUMI DOMENICALI





# INDIVIDUAZIONE DEL TREND

```
giorni = (linspace(1, 104, n))';  
phi = [ones(n, 1), giorni];  
[thetals, devthetals] = lscov(phi, consumi);  
giorni_3anni = (linspace(1, 156, n*3/2))';  
trend = thetals(1, :) + giorni_3anni.*thetals(2,:);  
plot(giorni_3anni, trend, 'k');
```



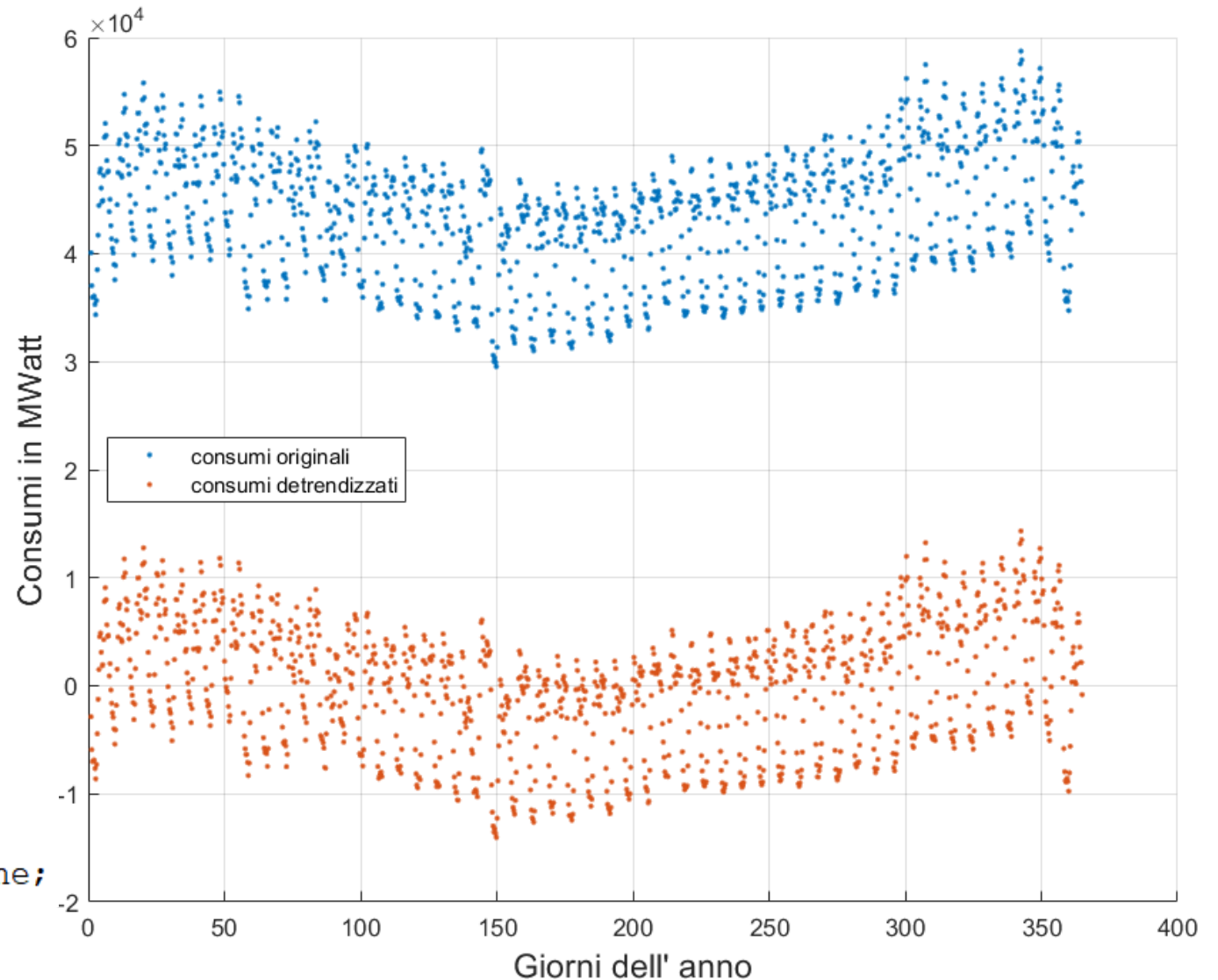
Disponendo di dati relativi ad un periodo di soli due anni, ci limitiamo a stimare il trend con un *modello lineare* di primo ordine.

# DETRENDIZZAZIONE

L'**identificazione** viene effettuata su dati detrendizzati rispetto all'anno di identificazione.

In fase di **validazione** la stima viene confrontata con i dati di validazione detrendizzati rispetto al trend dell'anno di validazione.

```
phil = [ones(n, 1), giorni];  
[thetals1, devthetals1] = lscov(phil, consumi);  
trend_identificazione = phil * thetals1;  
consumi_detrend = consumi - trend_identificazione;
```

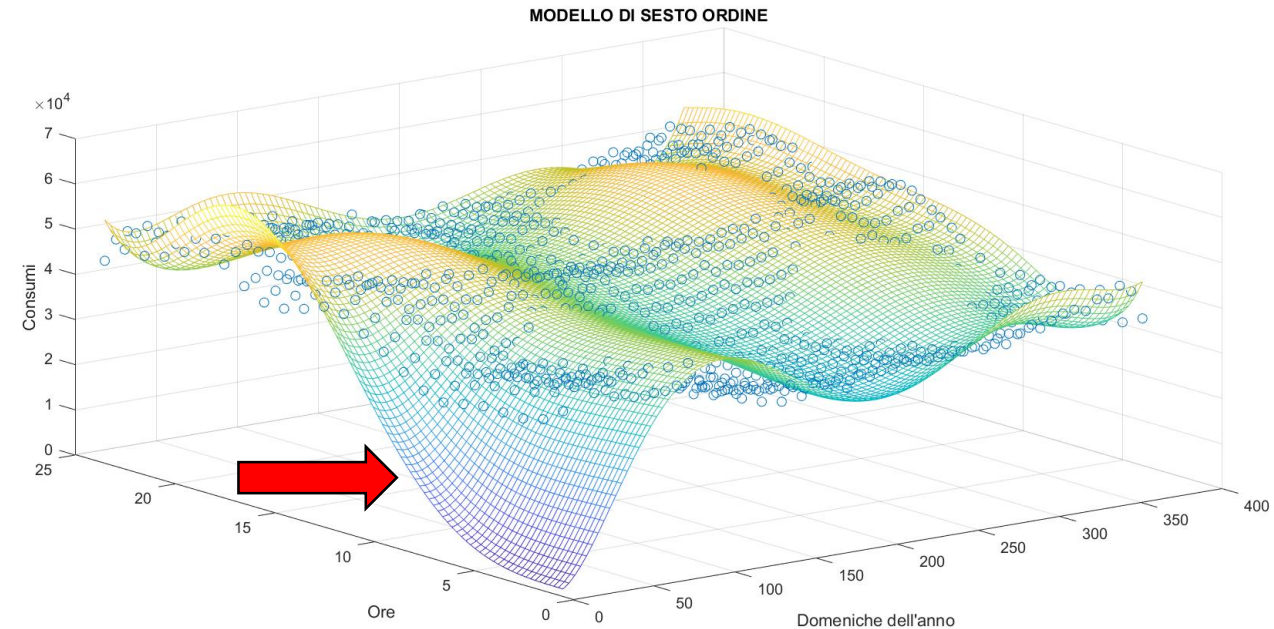
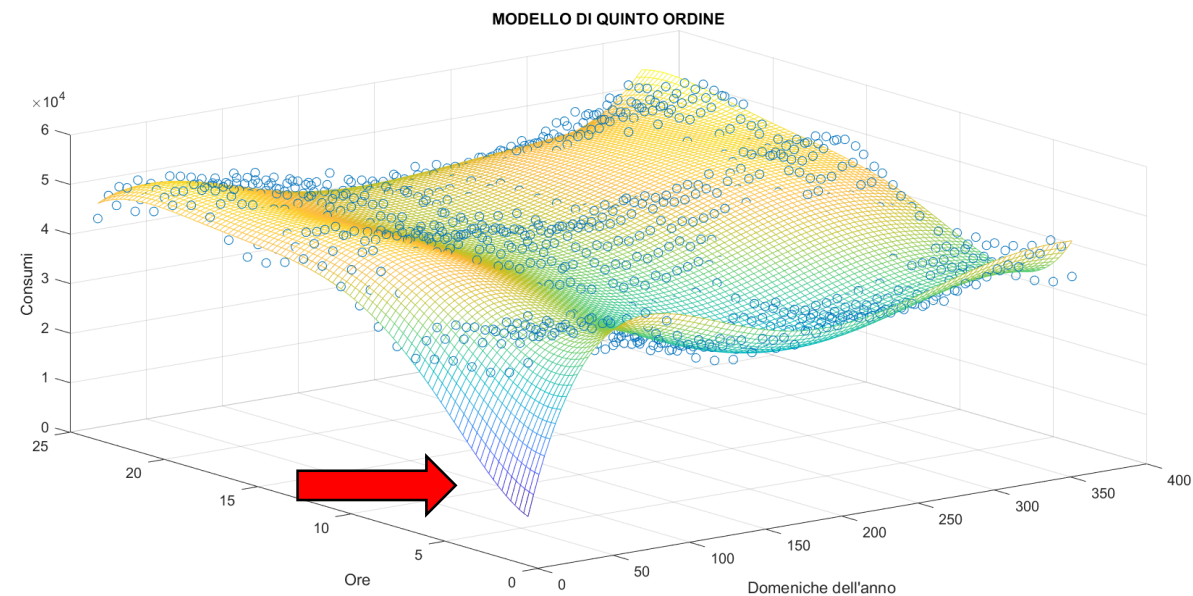


# PRIMO APPROCCIO: MODELLI POLINOMIALI

In prima analisi abbiamo utilizzato modelli polinomiali di vario ordine.



# MODELLI CON OVERFITTING



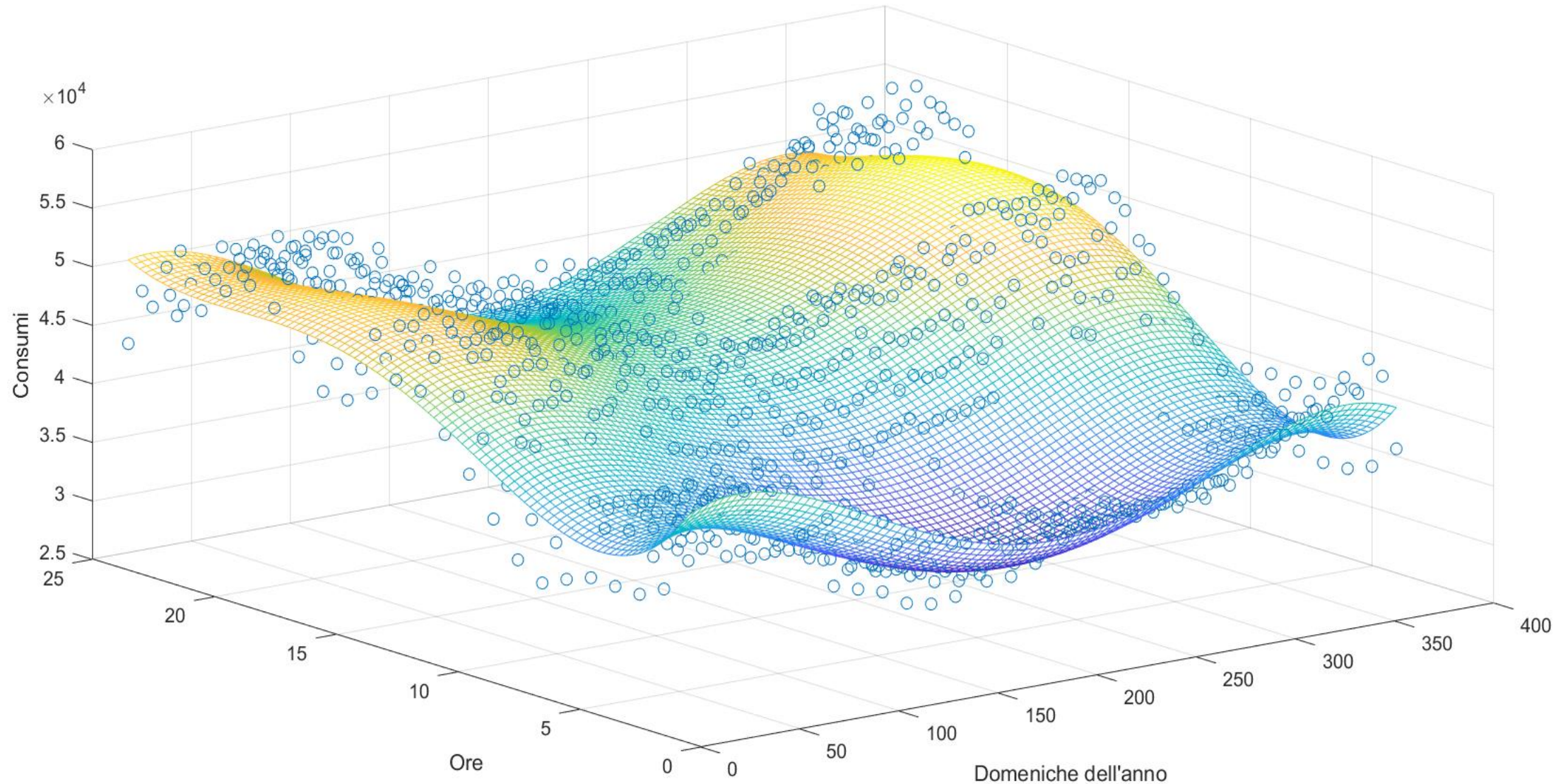
I modelli di ordine superiore al quarto tendono a overfittare i dati di identificazione



# MODELLO DEL 4° ORDINE

```
--  
phi4 = [ones(n,1),...  
        giorni, ore,...  
        giorni.^2, ore.^2, giorni.*ore, ...  
        giorni.^3, ore.^3, (giorni.^2).*ore, (ore.^2).*giorni, ...  
        giorni.^4, ore.^4, giorni.^3.*ore, ore.^3.*giorni, giorni.^2.*ore.^2];  
[thetals4, devthetals4] = lscov(phi4, consumi);  
stima_consumi4 = phi4*thetals4;
```

# PLOT MODELLO QUARTO ORDINE



## SECONDO APPROCCIO: MODELLI DI FOURIER

I modelli polinomiali sembrano non seguire in maniera efficace gli andamenti periodici dei consumi.

Si passa quindi a considerare modelli basati sulle *serie di Fourier*.





# MODELLO DI FOURIER

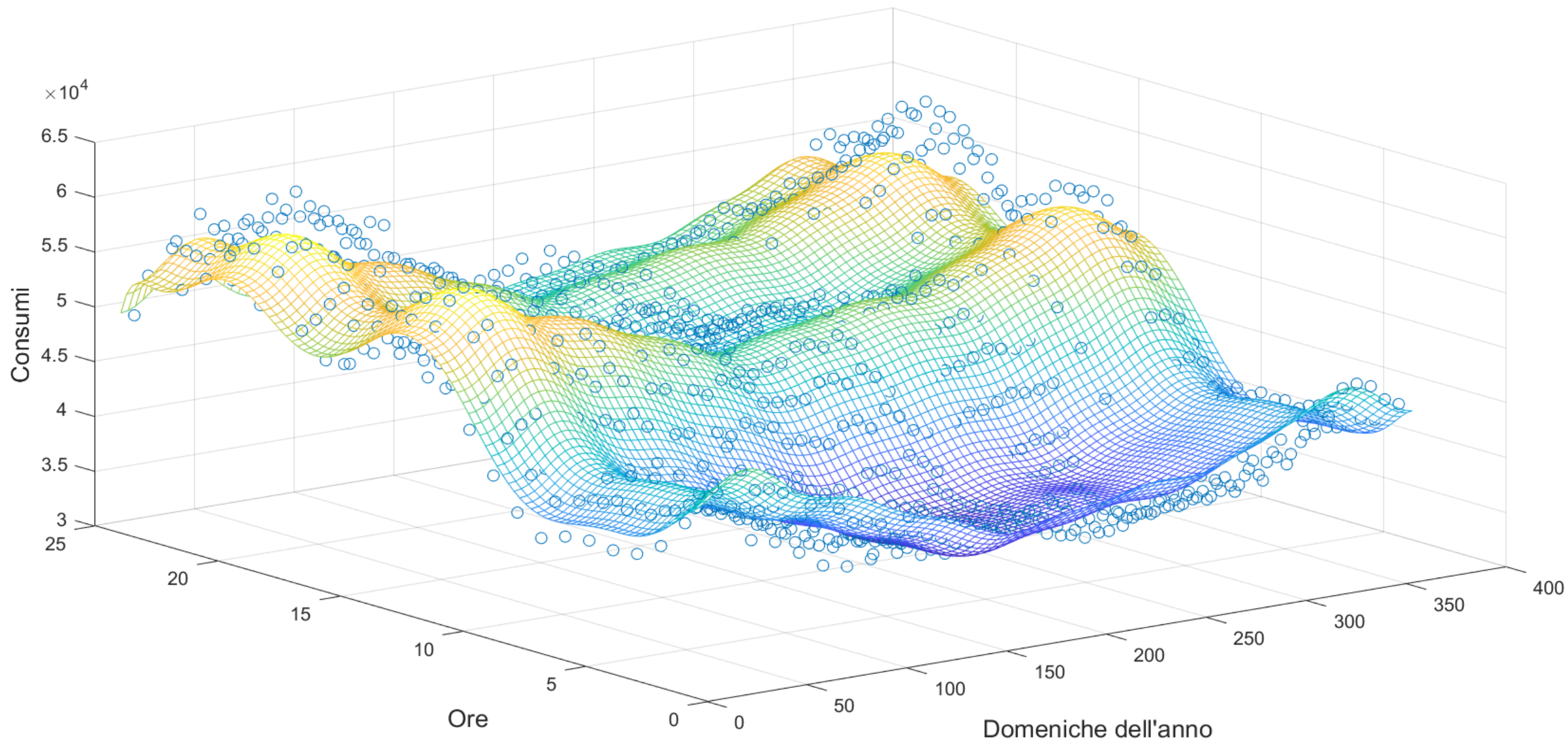
Dopo aver minimizzato il valore degli ssr di validazione al variare del numero di armoniche, il numero ottimo di armoniche risulta:

**8 armoniche per le ore, 7 armoniche per i giorni**

```
w1 = 2 * pi / 365;  
w2 = 2 * pi / 24;  
phiFourier = [cos(w1*giorni), sin(w1*giorni), cos(w2*ore), sin(w2*ore), ...  
              cos(2*w1*giorni), sin(2*w1*giorni), cos(2*w2*ore), sin(2*w2*ore), ...  
              cos(3*w1*giorni), sin(3*w1*giorni), cos(3*w2*ore), sin(3*w2*ore), ...  
              cos(4*w1*giorni), sin(4*w1*giorni), cos(4*w2*ore), sin(4*w2*ore), ...  
              cos(5*w1*giorni), sin(5*w1*giorni), cos(5*w2*ore), sin(5*w2*ore), ...  
              cos(6*w1*giorni), sin(6*w1*giorni), cos(6*w2*ore), sin(6*w2*ore), ...  
              cos(7*w1*giorni), sin(7*w1*giorni), cos(7*w2*ore), sin(7*w2*ore), ...  
              cos(8*w2*ore), sin(8*w2*ore)];  
[thetalsFourier, devthetalsFourier] = lsconv(phiFourier, consumi_detrendizzati);  
  
stima_consumiFourier = phiFourier * thetalsFourier + stima_trend;
```



# PLOT MODELLO DI FOURIER



# TERZO APPROCCIO: SOMMA DI MODELLI

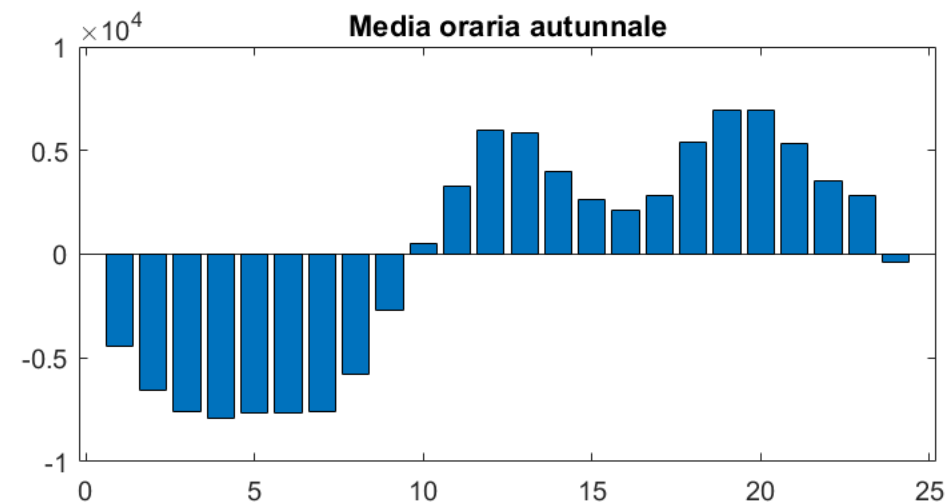
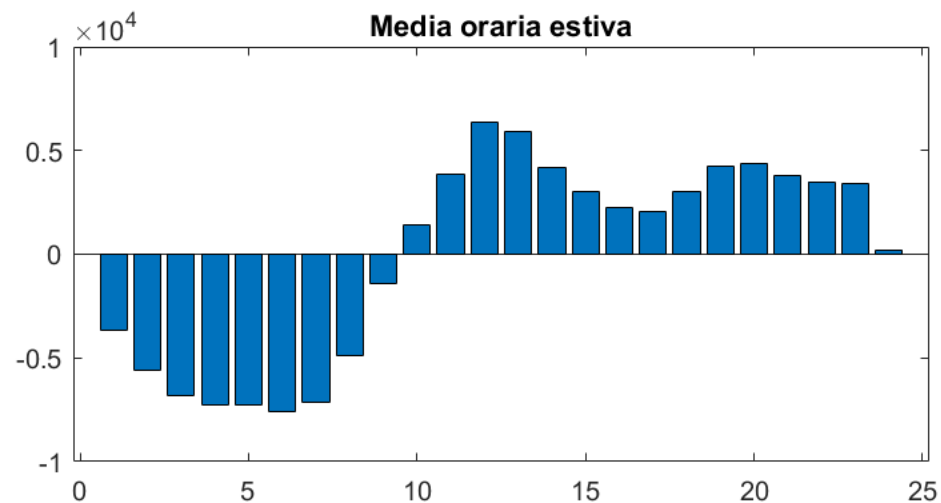
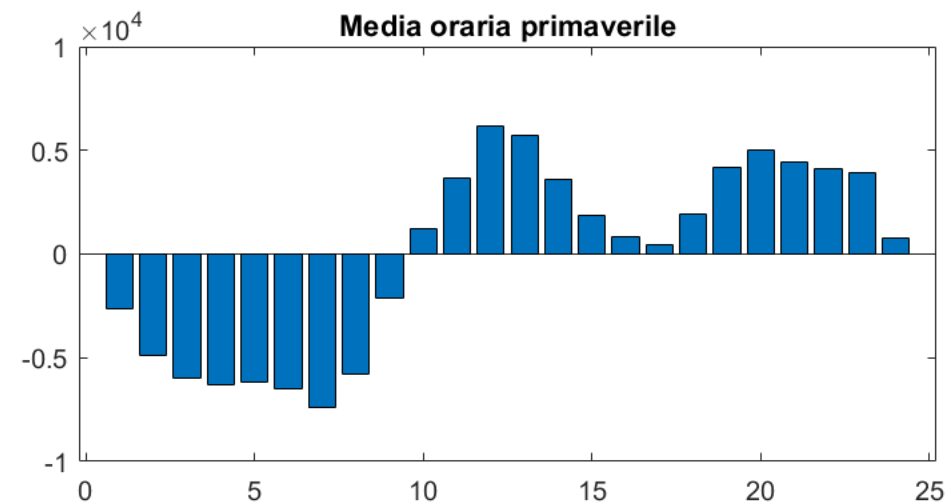
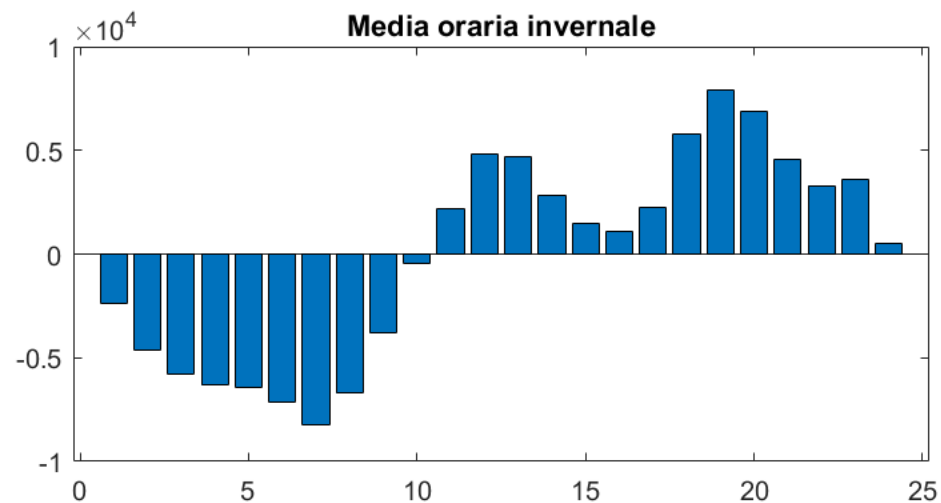
Abbiamo considerato un modello per l'andamento dei consumi durante le 24 ore della giornata e uno per l'andamento dei consumi durante le 52 domeniche dell'anno.

Il modello complessivo è ottenuto sommando questi ed aggiungendo il trend previsto.

## RISULTATO

L'ssr di validazione ottenuto con questo modello è migliore solo del 3% rispetto al modello di Fourier





## OSSERVAZIONE

Gli andamenti orari nell'arco di una giornata risultano essere *diversi* in base alla stagione. Considerare le stagioni potrebbe portare ad un miglioramento della precisione rispetto al modello precedente.

# QUARTO APPROCCIO: SOMMA MODELLI CON STAGIONALITÀ

L'andamento sulle 24 ore dei consumi è stato stimato separatamente per le quattro stagioni.

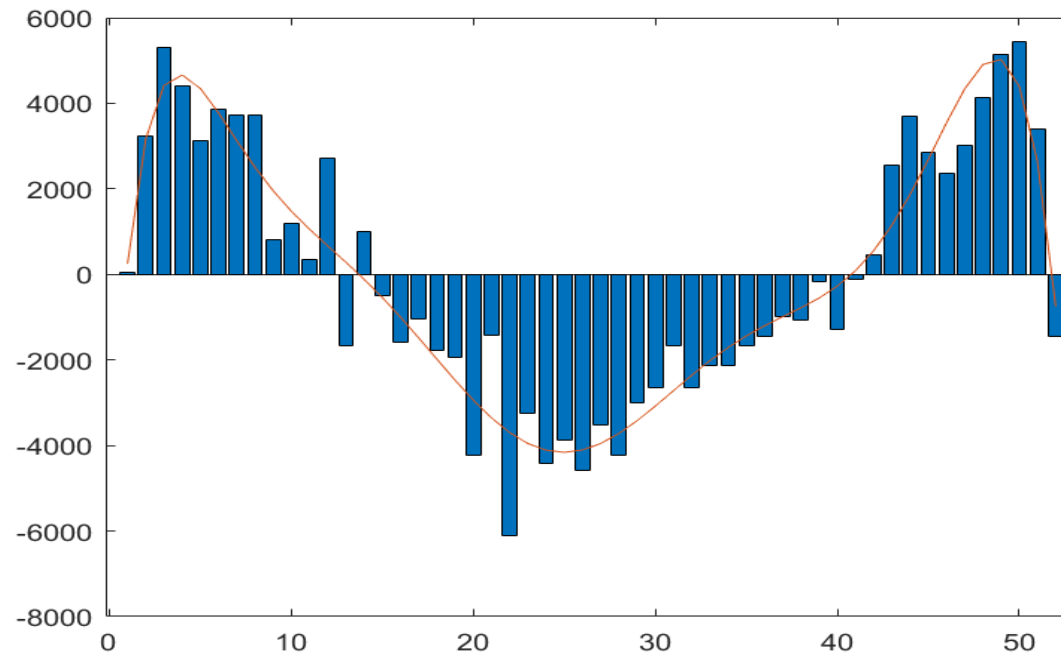
Il modello dei consumi giornalieri delle 52 domeniche rimane uguale a quello precedente.





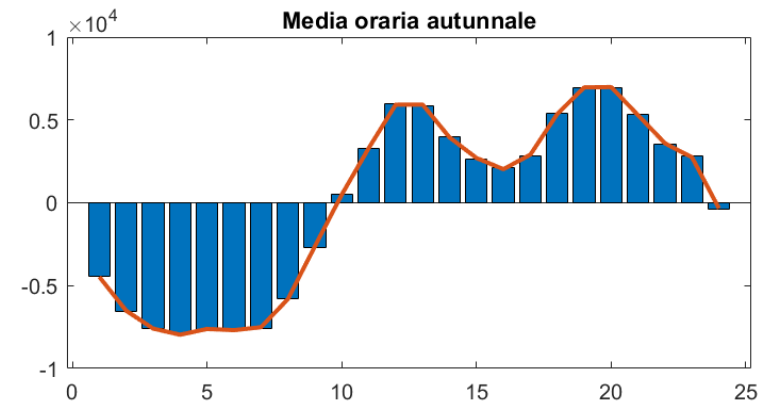
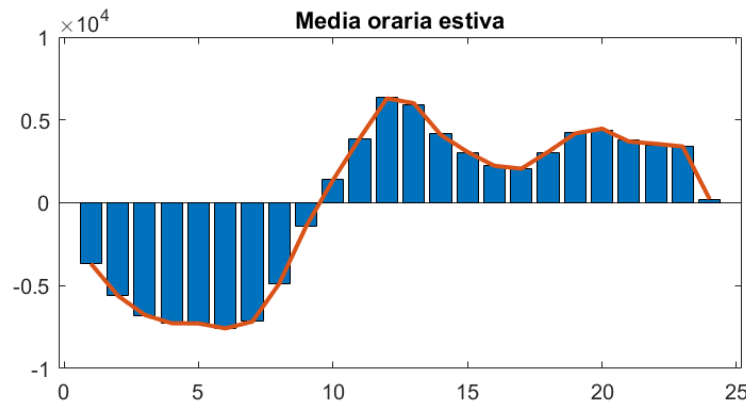
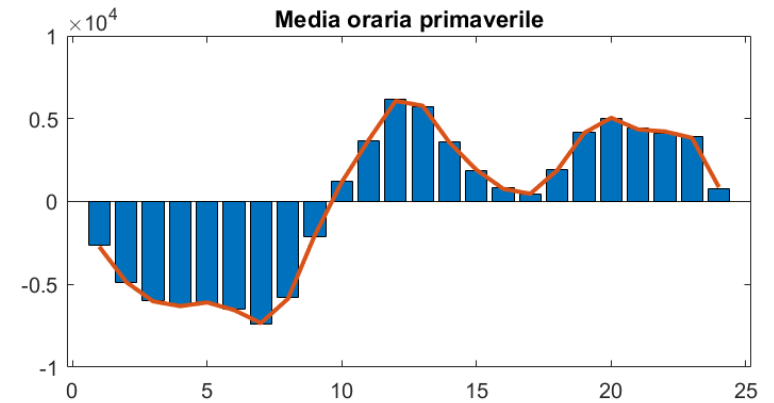
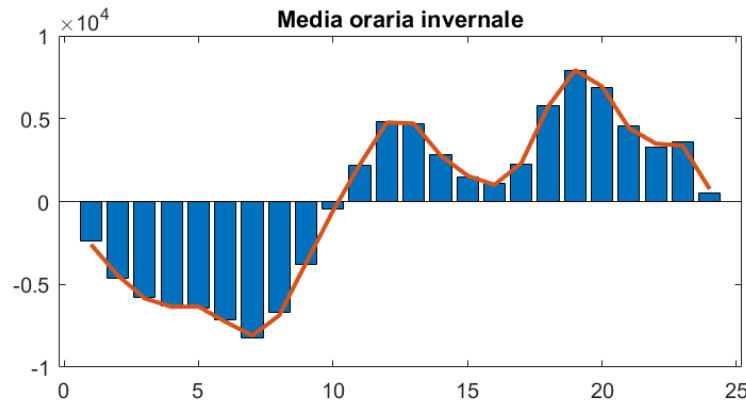
# MODELLO GIORNALIERO

```
w3 = 2 * pi / 365;  
phiFGiorni = [ cos(w3*domeniche'), sin(w3*domeniche'), ...  
               cos(2*w3*domeniche'), sin(2*w3*domeniche'), ...  
               cos(3*w3*domeniche'), sin(3*w3*domeniche'), ...  
               cos(4*w3*domeniche'), sin(4*w3*domeniche'), ...  
               cos(5*w3*domeniche'), sin(5*w3*domeniche')  
             ];
```



# MODELLO ORARIO

```
w2 = 2 * pi / 24;  
phiF = [ cos(w2*ore), sin(w2*ore),  
        cos(2*w2*ore), sin(2*w2*ore),  
        cos(3*w2*ore), sin(3*w2*ore),  
        cos(4*w2*ore), sin(4*w2*ore),  
        cos(5*w2*ore), sin(5*w2*ore),  
        cos(6*w2*ore), sin(6*w2*ore),  
        cos(7*w2*ore), sin(7*w2*ore),  
        cos(8*w2*ore), sin(8*w2*ore),  
        cos(9*w2*ore), sin(9*w2*ore)  
];
```



Si calcolano i quattro stimatori utilizzando la stessa **matrice di sensitività** (phiF) e per ciascuno i dati relativi alla stagione considerata.

## OSSERVAZIONE

Nella fase di individuazione del modello abbiamo scelto di identificare sul primo anno e di validare sul secondo. Invece, per la stima del terzo anno l'identificazione viene effettuata sul secondo anno piuttosto che sul primo o sulla media tra i due, poiché è più probabile che i consumi del terzo anno siano più simili a quelli del secondo rispetto che a quelli del primo.



# SCELTA DEL NUMERO DI ARMONICHE

Primo anno validazione, secondo anno identificazione

Armoniche ore	Armoniche giorni	SSR validazione
9	5	$3.5977 \cdot 10^9$
8	5	$3.6045 \cdot 10^9$
10	5	$3.5967 \cdot 10^9$
<b>11</b>	<b>5</b>	<b><math>3.5927 \cdot 10^9</math></b>
12	<b>5</b>	$3.5929 \cdot 10^9$
<b>11</b>	6	$3.5950 \cdot 10^9$
<b>11</b>	4	$3.8832 \cdot 10^9$

Primo anno identificazione, secondo anno validazione

Armoniche ore	Armoniche giorni	SSR validazione
10	6	$3.3668 \cdot 10^9$
10	<b>5</b>	$3.2715 \cdot 10^9$
10	4	$3.3077 \cdot 10^9$
<b>9</b>	<b>5</b>	<b><math>3.2710 \cdot 10^9</math></b>
11	6	$3.2783 \cdot 10^9$

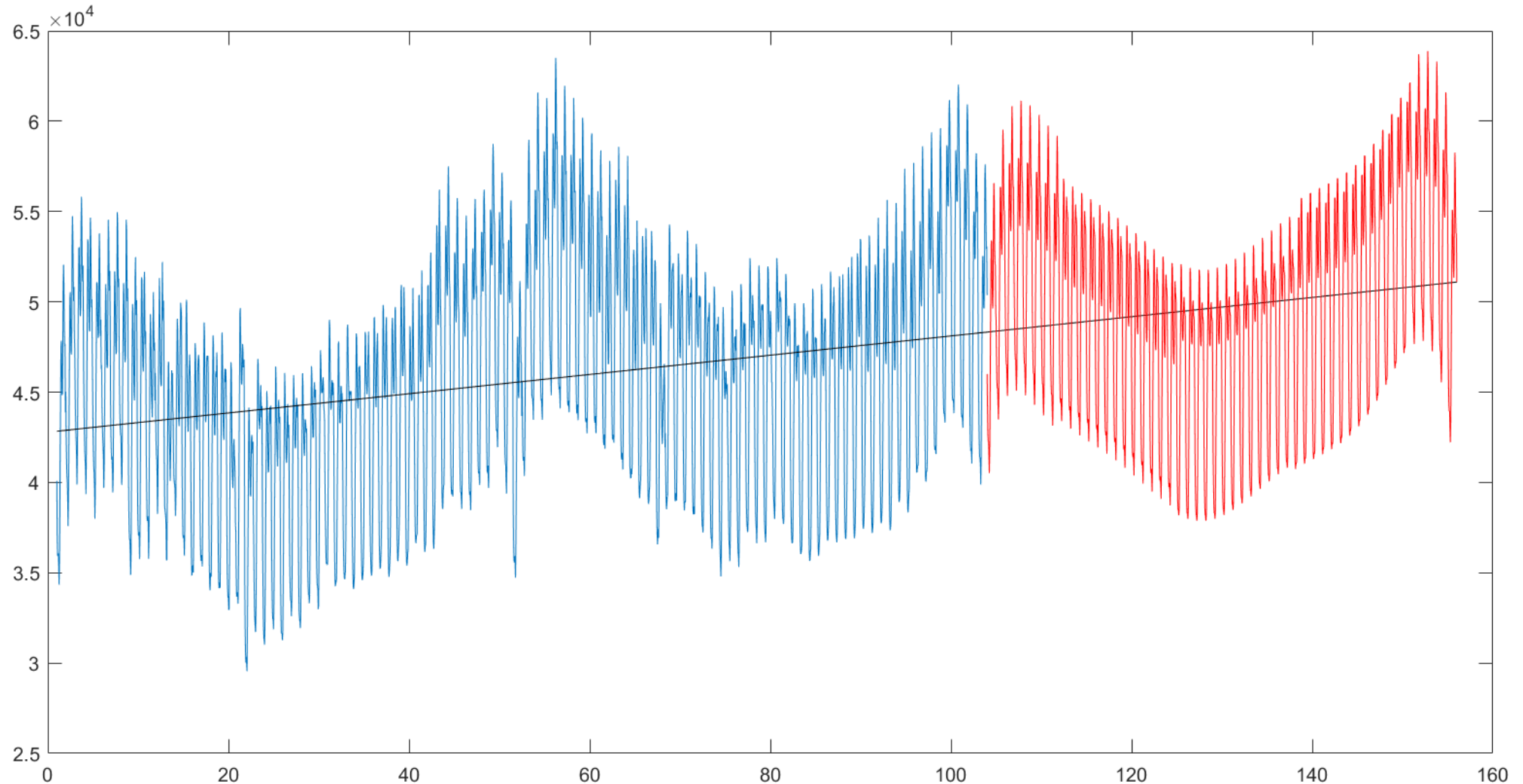
## OSSERVAZIONE

Nella prima tabella l'SSR aumenta di poco passando da 11 a 9 armoniche per le ore, si sceglie quindi questo secondo caso. Scelta finale: 9 armoniche per le ore, 5 per i giorni.



# STIMA

Come anticipato,  
si sommano i due modelli precedenti e, ad essi, si somma il trend previsto.



# INDICATORI DI PERFORMANCE

```
n= 52*24;
previsioneStagionaleVal = consumiDetrendModelStagionale + trendVal;
epsilonValStagionale = consumiDomenicaliVal - previsioneStagionaleVal;
ssrValStagionale = epsilonValStagionale' * epsilonValStagionale;
mseValStagionale = ssrValStagionale /n;
rmsdValStagionale = sqrt(mseValStagionale);
range = (max(consumiDomenicaliVal)-min(consumiDomenicaliVal));
nrmsd_mediaValStagionale = (rmsdValStagionale / range ) *100;
nrmsd_rangeValStagionale = (rmsdValStagionale / (mean(consumiDomenicaliVal))) *100;
maeValStagionale= mean(abs(epsilonValStagionale));
```

INDICATORE	VALORE
SSR	$3.27 \cdot 10^9$
MSE	$3.19 \cdot 10^6$
RMSD	$1.79 \cdot 10^3$
NRMSD (media)	6.23%
NRMSD (range)	3.64%
MAE	$1.29 \cdot 10^3$

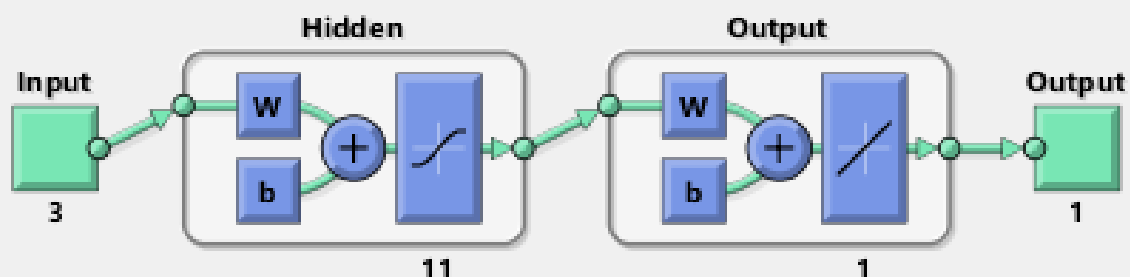
# MODELLO DEFINITIVO: IMPLEMENTAZIONE DI RETE NEURALE

Abbiamo utilizzato una rete neurale con uno strato nascosto da 11 neuroni, dando come dati di training i consumi del primo anno e come target i consumi del secondo anno, entrambi detrendizzati.

Non essendo disponibile un terzo anno su cui effettuare la validazione si è cercato un modello che minimizzasse l'SSR per i dati di target mantenendolo basso anche per i dati di training, in modo da non overfittare il target.

# RETE NEURALE E TRAINING

Neural Network



```
input = [consumi1,giorni1,ore1]';
```

```
net = feedforwardnet(11);
```

```
net.divideParam.trainRatio = 0.7;
```

```
net.divideParam.valRatio = 0.2;
```

```
net.divideParam.testRatio = 0.3;
```

```
[net,tr] = train(net,input,consumi2);
```

```
output = net(input);
```



# INDICATORI DI PERFORMANCE

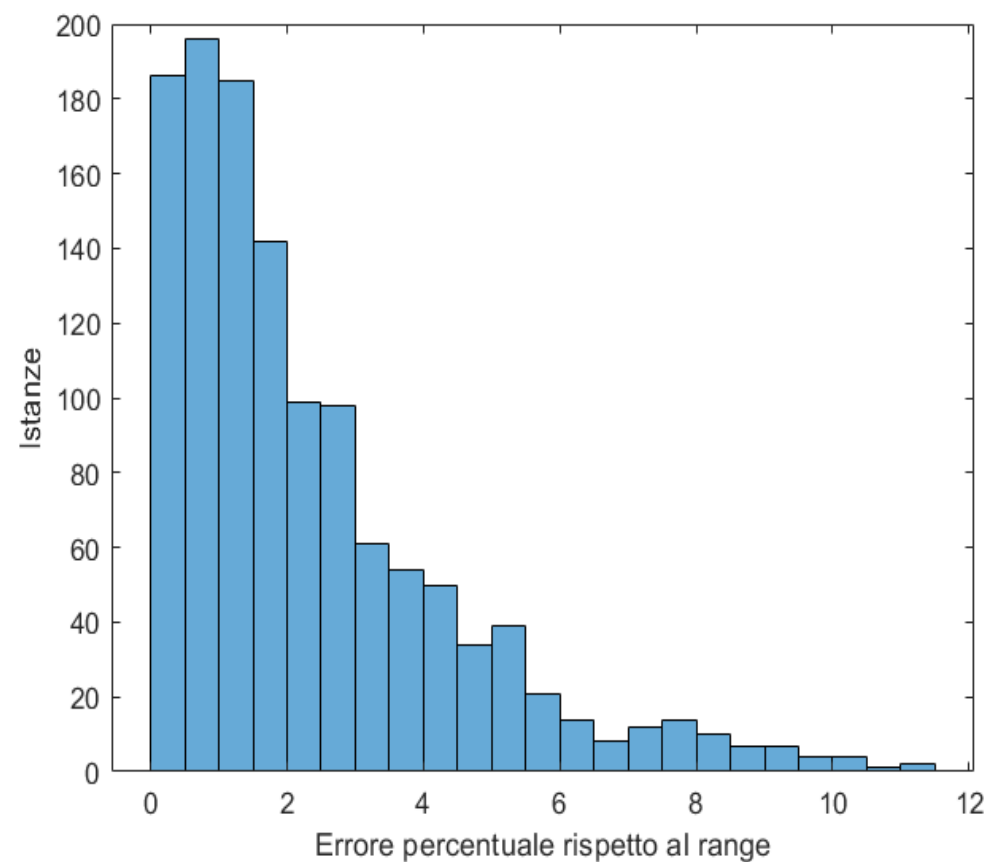
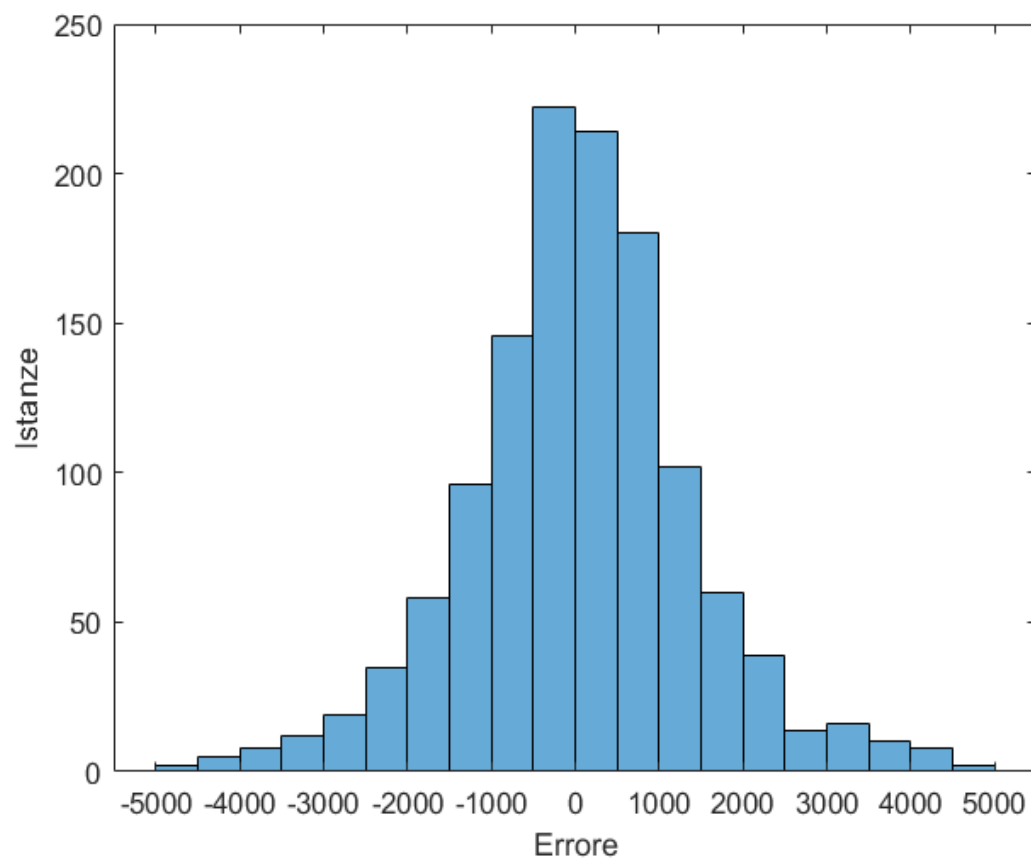
## TRAINING:

INDICATORE	VALORE
SSR	$2.28 \cdot 10^9$
MSE	$2.23 \cdot 10^6$
RMSE	$1.49 \cdot 10^3$
NRMSD (media)	3.41%
NRMSD (range)	5.11%
MAE	$1.04 \cdot 10^3$

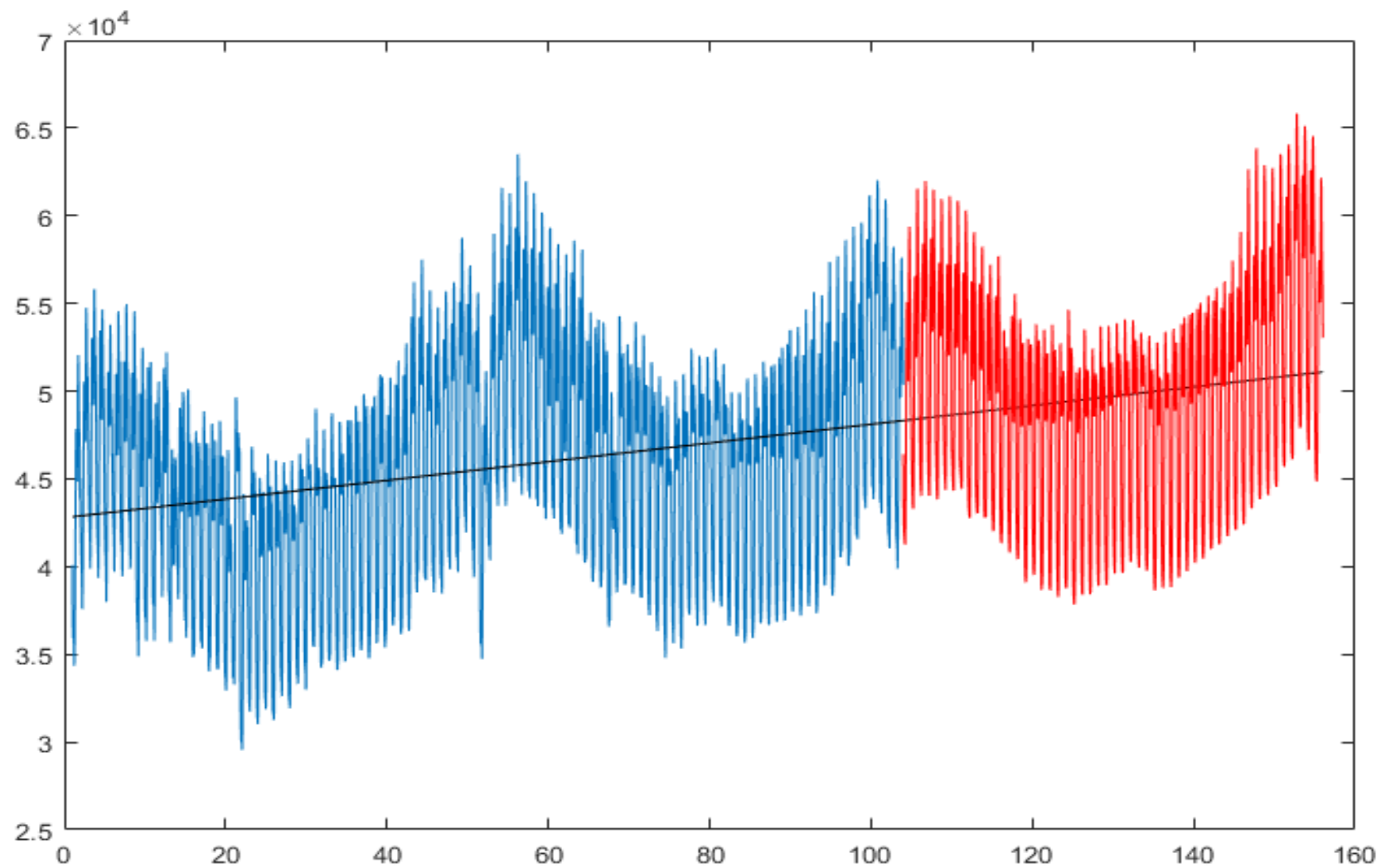
## TARGET:

INDICATORE	VALORE
SSR	$2.23 \cdot 10^9$
MSE	$2.18 \cdot 10^6$
RMSE	$1.48 \cdot 10^3$
NRMSD (media)	3.11%
NRMSD (range)	5.14%
MAE	$1.01 \cdot 10^3$

# ISTOGRAMMI DEGLI ERRORI



# STIMA FINALE



# CONFRONTO FINALE

MODELLO	SSR DI VALIDAZIONE
POLINOMIALE	$1.32 \cdot 10^{10}$
FOURIER	$4.06 \cdot 10^9$
SOMMA DI MODELLI (ORARIO + ANNUALE)	$3.93 \cdot 10^9$
SOMMA DI MODELLI CON STAGIONALITÀ	$3.27 \cdot 10^9$
<b>RETE NEURALE</b>	<b><math>2.28 \cdot 10^9</math></b>