
Table of Contents

VU Energiemodelle 2014 - Übung 1 - Regressionsanalyse - SS2014	1
Kurze Deskriptive Darstellung der Zusammenhänge:	1
Lineare Regression zur Abschätzung der beobachteten Strompreise	3
KOMMENTAR!!!	7
Hier wird noch kurz ein Ansatz mit LAG angedeutet	8
Kommentar	10

VU Energiemodelle 2014 - Übung 1 - Regressionsanalyse - SS2014

Michael Hartner

```
% Dieses Beispiel zur Abschätzung von Einflüssen auf den Strompreis die als  
% Hilfe zur Umsetzung der Übungsaufgaben! Es handelt sich um sehr  
% vereinfachte Ansätze, die hauptsächlich die Umsetzung der linearen  
% Regression in Matlab illustrieren sollen.
```

```
clc % löscht Command Window  
clear all % löscht alle Variablen aus Workspace  
close all % Schließt alle geöffneten Abbildungen
```

```
data_price=dataset('XLSFile','Preise_import.xlsx');  
% Das File Preise_import.xlsx entspricht dem File Spotmarktpreise_2012.xlsx  
% Es wurden nur die Spaltennamen für den Import vereinfacht.
```

Kurze Deskriptive Darstellung der Zusammenhänge:

Hier werden kurz die empirischen Daten dargestellt

```
% Aus diesem Plot lässt sich eine positive Korrelation zwischen Preisen und  
% Last vermuten. Aus Energieökonomie sollte dieser Zusammenhang bekannt und  
% auch erklärbar sein!!! Erinnern Sie sich an die Merit Order der  
% Kraftwerke und die Annahme, dass sich der Preis aus den Grenzkosten des  
% teuersten Kraftwerks bildet!
```

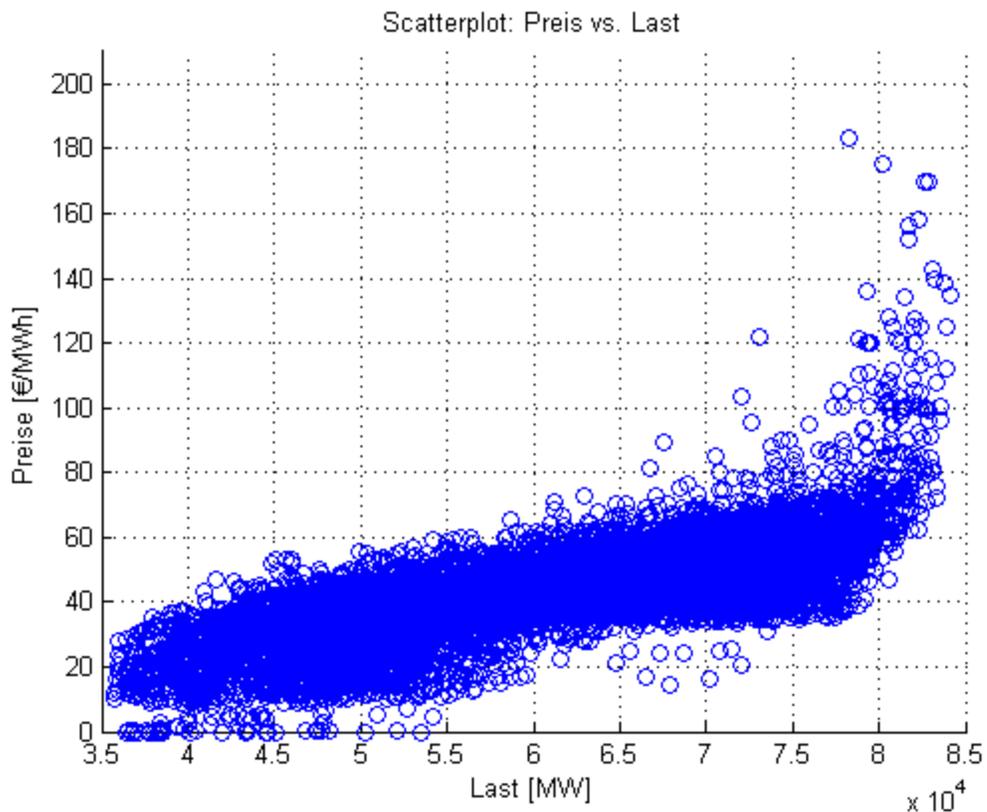
```
scatter(data_price.Last,data_price.Preis);  
ylim([0 max(data_price.Preis)]) % die negativen Strompreise werden nicht in  
% der Abbildung gezeigt und auch nicht diskutiert.  
title('Scatterplot: Preis vs. Last')  
xlabel('Last [MW]')  
ylabel('Preise [€/MWh]')  
grid on
```

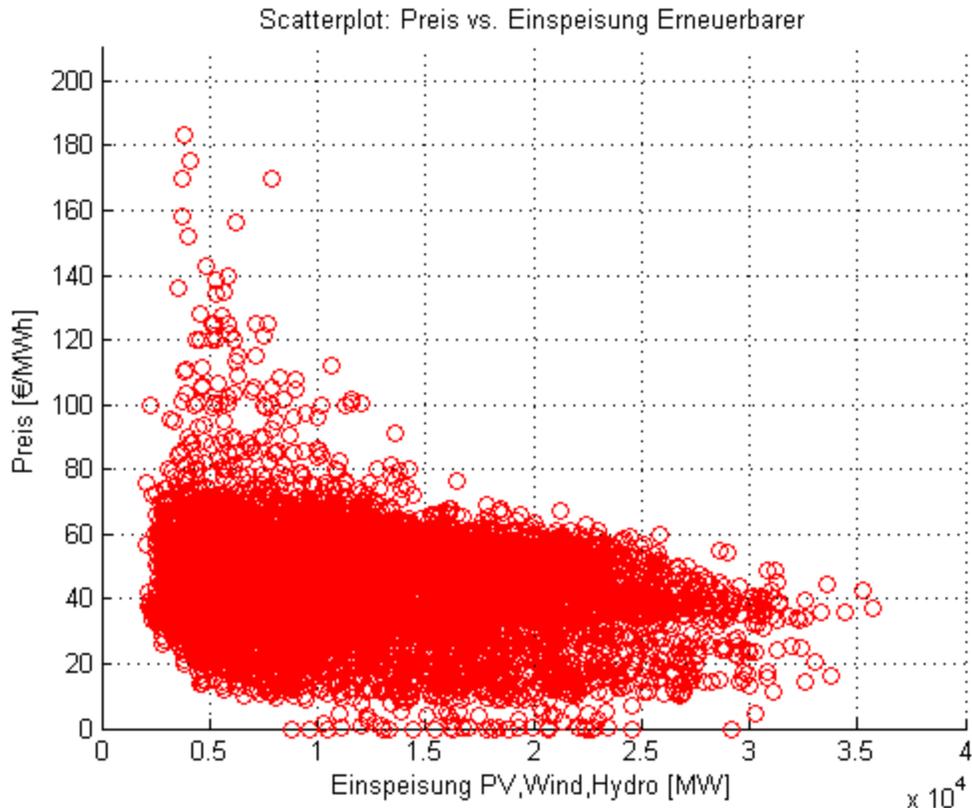
```
% In diesem Plot wird die empirische Korrelation zwischen der Einspeisung  
% erneuerbarer Energie (mit Grenzkosten nahe 0) und dem Strompreis. Hier
```

```
% ist eine negative Korrelation zu erwarten, da ja die Einspeisung
% dieser Technologien die Angebotskurve nach rechts verschiebt und damit
% Kraftwerke mit geringeren Grenzkosten zum Einsatz kommen.
```

```
figure
scatter(data_price.RES,data_price.Preis,'r');
ylim([0 max(data_price.Preis)])
xlabel('Einspeisung PV,Wind,Hydro [MW]')
ylabel('Preis [€/MWh]')
title('Scatterplot: Preis vs. Einspeisung Erneuerbarer')
grid on
```

```
% mögliche interessante Darstellung wären auch die Dauerlinien der Last,
% Residuallast und der Strompreise. Diese werden hier allerdings nicht
% gezeigt, da der Fokus auf der linearen Regression liegt!!!
```





Lineare Regression zur Abschätzung der beobachteten Strompreise

```

fprintf('MODELLANSATZ 1:')
fprintf('\n')
% Durch die Methodik der linearen Regression können beide Einflussfaktoren
% (Last, Einspeisung) abgebildet werden.

% Hier werden 2 Modellansätze gezeigt. Natürlich gibt es noch viele weitere
% Ansätze und auch noch einige Einflussfaktoren, die hier nicht gezeigt
% werden.

% Hier wird der Befehl "fitlm" gewählt um die Regression durchzuführen. Für
% mehr Infos dazu geben sie einfach "doc fitlm" im Command Window ein um
% zur Hilfe zu gelangen.

% Erstellung der Inputdaten für das Modell:
input_1=[data_price.RES,data_price.Last]; % dieser Befehl erzeugt eine Matrix aus
% den Vektoren RES und Last. Diese werden dem Modell als unabhängige
% Inputvariable übergeben. Die Konstante wird in dem Befehl "fitlm" selbst
% hinzugefügt - es reicht also nur die Inputs ohne Konstante zu übergeben.
% Die Konstante entspricht einfach einem Vektor der Länge der
% Beobachtungen und nur "1" als Einträge. Der Befehl "fitlm" schätzt dazu
% direkt den Koeffizienten beta0, der als Intercept(=Schnittpunkt)

```

```

% ausgegeben wird.

% Annahme für den Zusammenhang:  $p(t)=b_0+b_1 \cdot \text{RES}(t)+b_2 \cdot \text{Last}(t)$ 
% Man geht also von einem linearen Zusammenhang aus.

Modell_1=fitlm(input_1,data_price.Preis,'linear'); % Mit diesem Befehl wird
% die Regression ausgeführt. Die Ergebnisse werden in dem Objekt "Modell_1"
% gespeichert. Siehe Workspace. Über Doppelklick können Sie sich den Inhalt
% ansehen. Für Sie sind vor allem die Variablen "Modell_1.Coefficients" sowie
% "Modell_1.Rsquared" interessant. Mehr dazu erfahren Sie ebenfalls in der
% Hilfe!

%  $y = b_0 \cdot 1 + b_1 \cdot x_1 + b_2 \cdot x_2$  - so sieht das geschätzte Modell aus, wobei
% Sie sich  $y(\text{Preis})$   $x_1(\text{RES})$  und  $x_2(\text{Last})$  also Vektoren mit den jeweiligen
% Beobachtung vorstellen müssen.

disp(Modell_1); % Damit werden die wichtigsten Ergebnisse im Command Window
% angezeigt. Hier finden Sie sowohl die Koeffizienten ( $b_0, b_1, b_2$ ) unter
% Spalte "Estimate" sowie alle weiteren für die Übung Relevanten
% Statistiken.

% Vergleich von Modell und Messungen für einen bestimmten
% Beobachtungszeitraum:

t=[96:192,4080:4176]'; % damit werden jeweils 4 Tage im Winter und 4 Tage im
% Sommer ausgewählt - Beachten Sie, dass diese im anschließenden Diagramm
% einfach aneinander gereiht werden

% Hier werden zwei Varianten gezeigt:

% Variante 1: hier wird nur einmal gezeigt, wie Sie aus den berechneten
% Koeffizient einfach durch Einsetzen, die vom Modell abgeschätzten Werte
% für die Abhängige Variable berechnen können. Dies dient zur
% Veranschaulichung - Matlab stellt hier die Funktion "predict" zur
% Verfügung, was Ihnen Schreibearbeit ersparen kann!

% Über den zuvor erzeugten Index t wählen Sie die gesuchten Beobachtungen
% aus.

mod_price_var1=Modell_1.Coefficients.Estimate(1)+...
    Modell_1.Coefficients.Estimate(2)*data_price.RES(t)+...
    Modell_1.Coefficients.Estimate(3)*data_price.Last(t);

% Variante 2: Berechnung über Matlab-Funktion "predict". Siehe Hilfe mit
% "doc predict"!!!! Variante 1 und 2 liefern die gleichen Ergebnisse!!!

mod_price_var2=predict(Modell_1,input_1(t,:));

% Grafischer Vergleich:

figure
plot(data_price.Preis(t)) % auch hier werden die dazugehörigen Beobachtung
% für die abhängige Variable über den Index t ausgewählt.

```

```

hold on
plot(mod_price_var2,'r')
legend('Historische Preise', 'Modellpreise linear')
xlabel('Stunden des Betrachtungszeitraums')
ylabel('Preis [€/MWh]')
title('Vergleich: Modell und Beobachtungen')

```

MODELLANSATZ 1:

Linear regression model:

$$y \sim 1 + x1 + x2$$

Estimated Coefficients:

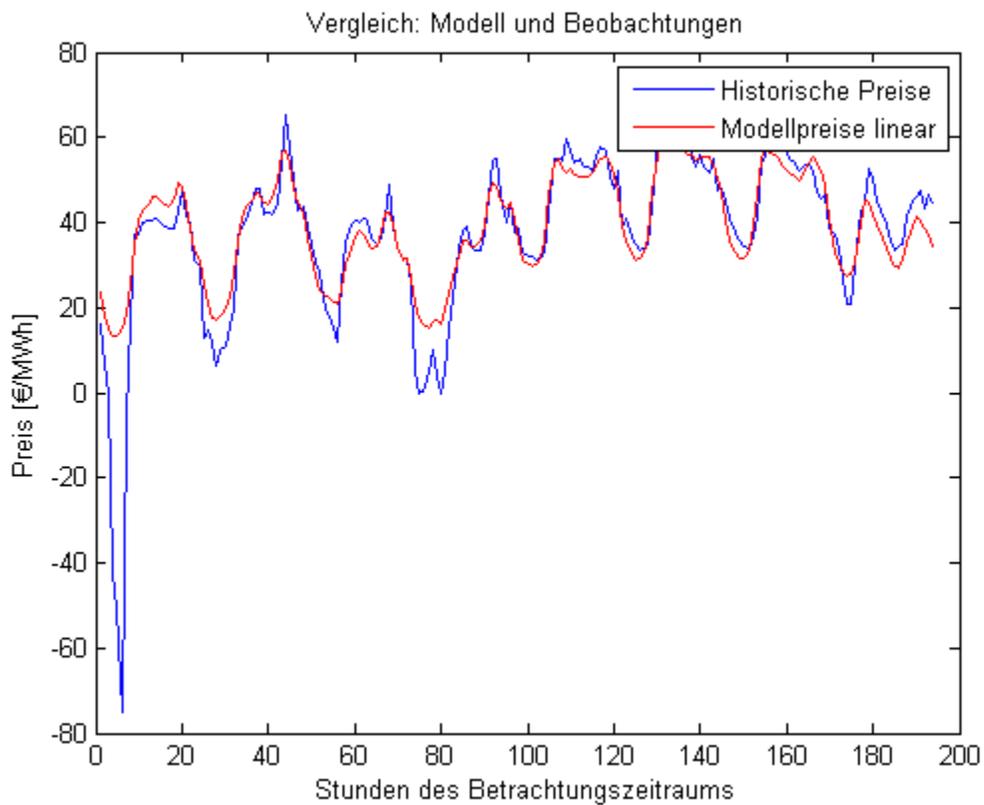
	Estimate	SE	tStat	pValue
(Intercept)	-20.117	0.65074	-30.914	2.7608e-199
x1	-0.0011411	1.9921e-05	-57.281	0
x2	0.0012561	1.0836e-05	115.92	0

Number of observations: 8784, Error degrees of freedom: 8781

Root Mean Squared Error: 11.5

R-squared: 0.621, Adjusted R-Squared 0.621

F-statistic vs. constant model: 7.21e+03, p-value = 0



```

fprintf('MODELLANSATZ 2:')
fprintf('\n')

```

```

% Hier wird noch ein zweiter Modellansatz gezeigt. Die Vorgehensweise ist
% die gleiche, die Inputdaten werden allerdings zuvor manipuliert.

% Annahme für den Zusammenhang: p(t)=K*RES(t)^b1*Last(t)^b2
% In dieser Form kann keine LINEARE Regression durchgeführt werden. Die
% Daten müssen zuvor logarithmiert werden.

input_2=[log(data_price.RES),log(data_price.Last)];

% Um das Modell rechnen zu können, müssen zunächst alle Preise <= 0
% entfernt werden - hier werden Sie einfach auf 0.001 gesetzt. Wenn der
% Anteil negativer Preise höher wäre, wäre dies natürlich keine zulässige
% Vorgehensweise!!!

price_neu=data_price.Preis; % Zuweisen auf andere Variable

price_neu(price_neu<=0)=0.001; % Manipulation der Preise <=0

Modell_2=fitlm(input_2,log(price_neu),'linear'); % auch die
% Abhängige wird hier logarithmiert!

disp(Modell_2)

% Natürlich muss hier wieder der Logarithmus aufgelöst werden
% (Befehl "exp") um die modellierten Preise zu berechnen.
mod_price_modell2=exp(predict(Modell_2,input_2(t,:)));

plot(mod_price_modell2,'green')
legend('Historische Preise', 'Modellpreise linear', 'Modellpreise log')

```

MODELLANSATZ 2:

Linear regression model:

$$y \sim 1 + x1 + x2$$

Estimated Coefficients:

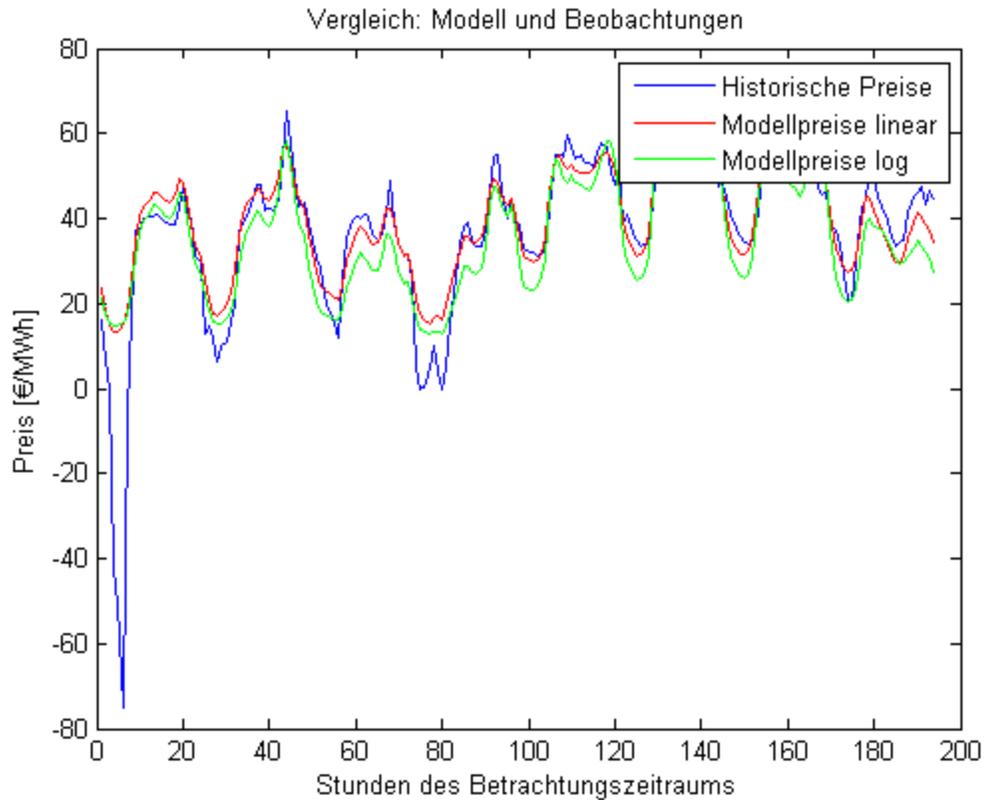
	Estimate	SE	tStat	pValue
(Intercept)	-20.015	0.48722	-41.081	0
x1	-0.50108	0.015942	-31.432	1.343e-205
x2	2.5698	0.045388	56.619	0

Number of observations: 8784, Error degrees of freedom: 8781

Root Mean Squared Error: 0.828

R-squared: 0.289, Adjusted R-Squared 0.289

F-statistic vs. constant model: 1.79e+03, p-value = 0



KOMMENTAR!!!

```
% der Modellansatz 2 muss in dieser Form jedenfalls verworfen werden, auch
% wenn die Ergebnisse im historischen Vergleich zu stimmen scheinen. Für
% Betrachtungszeiträume mit sehr hoher Nachfrage liefert das Modell
% allerdings viel zu hohe Preise. Sehen Sie sich die Ergebnisse noch einmal
% für Zeitraum November an!
```

```
% Versuchen Sie zu interpretieren warum das so ist.
% Wie sieht der Zusammenhang zwischen Preise und Last bzw. RES in
% den beiden Modellen aus?
```

Hier noch ein Darstellung in der die Inputfaktoren gemeinsam mit dem Modelloutput verglichen werden

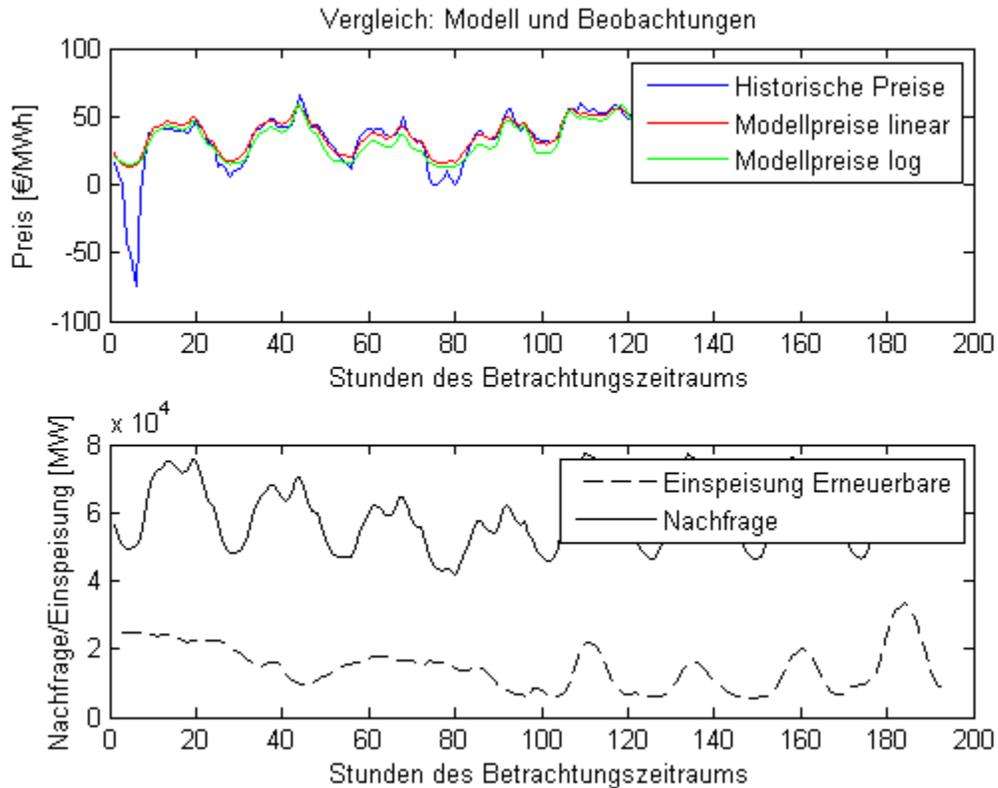
```
figure
subplot(2,1,1)
plot(data_price.Preis(t))
hold on
plot(mod_price_var2,'r')
plot(mod_price_modell2,'green')
legend('Historische Preise', 'Modellpreise linear', 'Modellpreise log')
xlabel('Stunden des Betrachtungszeitraums')
ylabel('Preis [€/MWh]')
title('Vergleich: Modell und Beobachtungen')

subplot(2,1,2)
```

```

plot(data_price.RES(t), 'Color', 'black', 'LineStyle', '--')
hold on
plot(data_price.Last(t), 'black')
legend('Einspeisung Erneuerbare', 'Nachfrage')
xlabel('Stunden des Betrachtungszeitraums')
ylabel('Nachfrage/Einspeisung [MW]')

```



Hier wird noch kurz ein Ansatz mit LAG angedeutet

Die Preise in einem Zeitpunkt hängen also von den Preisen im vorhergehenden Zeitpunkt ab.

```

fprintf('MODELLANSATZ 2: mit LAG')
fprintf('\n')

```

```

% Hier werden die Beobachtungen erst ab Zeitschritt 2 betrachtet. Der Preis
% der Vorperiode geht auch als Einflussvariable ein. Der Preis der letzten
% Beobachtung (end-1) natürlich nicht mehr. Hier müssen die Dimensionen der
% Vektoren übereinstimmen sonst gibt Matlab eine Fehlermeldung aus!!!
input_3=...
[data_price.RES(2:end),data_price.Last(2:end),data_price.Preis(1:end-1)];

```

```
Modell_3=fitlm(input_3,data_price.Preis(2:end),'linear'); % auch die
% Abhängige geht erst ab Beobachtung 2 in das Modell ein!!!
```

```
disp(Modell_3)
```

```
mod_price_modell3=predict(Modell_3,input_3(t-1,:));% t-1 weil die
% Beobachtungen um einen Zeitschritt verschoben sind.
```

```
figure
plot(data_price.Preis(t))
hold on
plot(mod_price_modell3,'green')
legend('Historische Preise', 'Modellpreise mit Lag')
```

MODELLANSATZ 2: mit LAG

Linear regression model:
 $y \sim 1 + x1 + x2 + x3$

Estimated Coefficients:

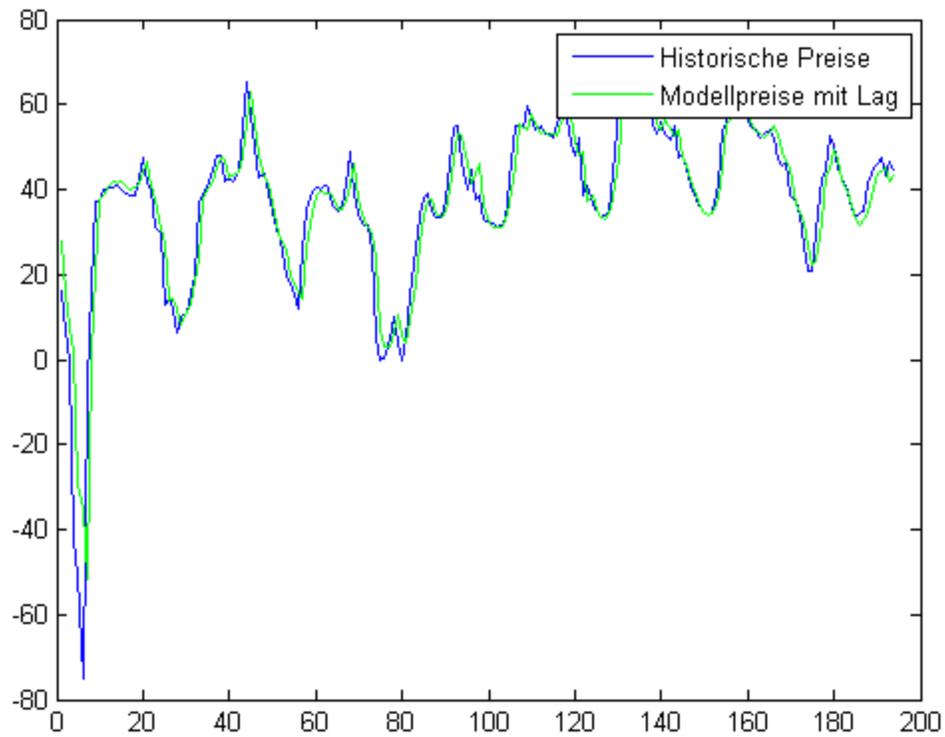
	<i>Estimate</i>	<i>SE</i>	<i>tStat</i>	<i>pValue</i>
<i>(Intercept)</i>	-7.3091	0.40417	-18.084	8.4348e-72
<i>x1</i>	-0.00036837	1.3475e-05	-27.338	5.4169e-158
<i>x2</i>	0.00037597	9.5998e-06	39.165	2.3299e-309
<i>x3</i>	0.73804	0.0059169	124.73	0

Number of observations: 8783, Error degrees of freedom: 8779

Root Mean Squared Error: 6.91

R-squared: 0.863, Adjusted R-Squared 0.863

F-statistic vs. constant model: 1.85e+04, p-value = 0



Kommentar

Auch wenn das Modell gute Ergebnisse liefert ist der Anwendungsbereich doch sehr eingeschränkt - für Strompreisprognosen für den Folgetag ist es beispielsweise ungeeignet, da das Ergebnis für die folgenden Stunden sehr stark vom letzten beobachteten Wert abhängt, was aber nicht unbedingt der Fall sein muss!

Published with MATLAB® R2013b