Universität Duisburg-Essen
Lehrstuhl für Ökonometrie
Prof. Dr. Christoph Hanck
M.Sc. Karolina Gliszczynska
Wintersemester 2022/2023

Methoden der Ökonometrie - Übung 3

Aufgabe 1

Öffnen Sie den Datensatz `USCrude` von US-Ölfeldern aus dem `AER` -Paket. Die Variable `price` gibt den Rohölpreis je Barrol an, `gravity` gibt die Dichte des Rohöls an (in API) und `sulphur` den Schwefelgehalt in Prozent.

1. Schätzen Sie die multiple Regression $\texttt{price} = \beta_1 + \beta_2\texttt{gravity} + \beta_3\texttt{sulphur} + u$
2. Schätzen Sie die Regression $\texttt{gravity} = \alpha_0 + \alpha_1\texttt{sulphur} + v$ und berechnen Sie die Residuen $\hat{v}$. Führen Sie jetzt eine Regression $\texttt{price} = \gamma_1\hat{v} + u$ durch. Was fällt auf?
3. Schätzen Sie das Modell $\texttt{price} = \phi_1 + \phi_2\texttt{sulphur} + w$ und berechnen Sie die Residuen $\hat{w}$. Regressieren Sie jetzt die Residuen $\hat{w}$ auf die Residuen $\hat{v}$ aus Aufgabenteil b). Was fällt auf? Vergleichen Sie die Residuen der jeweils letzten Regression der Aufgabenteile.

Aufgabe 2

Erzeugen Sie künstliche Daten mit `x <- 1:20` und `y <- x+rnorm(20)`. Fitten Sie ein Polynom in x zur Vorhersage von y. Berechnen Sie dazu $\hat{\boldsymbol{\beta}}$ indem Sie 1.) `lm()` nutzen und 2.) durch direkte Berechnung. Bei welchem Grad des Polynoms scheitert die direkte Berechnung? **Hinweis** Beachten Sie, dass die Funktion `I()` für die Formel der Polynome nutzen müssen, z.B. in `lm(y~x+I(x^2))`.

Aufgabe 3

Die direkte Berechnung von $\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y}$ ist ineffizient und instabil. Deshalb wird in der Funktion `lm()` zur Schätzung der Koeffizienten eine sogenannte QR-Zerlegung durchgeführt. Dabei wird die Matrix $\boldsymbol{X}, (n \times k)$, mit $rk(X) = k$ in eine orthogonale Matrix $\boldsymbol{Q}, (n \times k)$, (mit $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$) und eine invertierbare obere Dreiecksmatrix $\boldsymbol{R}, (k \times k)$, zerlegt: $\boldsymbol{QR} = \boldsymbol{X}$. Vergewissern Sie sich, dass $\boldsymbol{Q}^T\boldsymbol{y} = \boldsymbol{R}\hat{\boldsymbol{\beta}}$. Der numerische Vorteil ist folgender: Sobald man die QR-Zerlegung bestimmt hat, muss man nur noch eine Matrix mit einem Vektor multiplizieren ($\boldsymbol{Q}^T\boldsymbol{y} = \boldsymbol{q}$) und ein lineares Gleichungssystem mit einer oberen Dreiecksmatrix lösen ($\boldsymbol{R}\hat{\boldsymbol{\beta}} = \boldsymbol{q}$), was rekursiv mit der Funktion `backsolve()` in R möglich. Man muss also insbesondere nicht mehrere Matrixmultiplikationen durchführen und das Finden der Inverse entfällt, was numerisch häufig problematisch ist. Siehe dazu Aufgabe 2.
Schreiben Sie jetzt eine Funktion `my_lm()`, die zu einem gegeben Datensatz eine Konstante hinzufügt, mit `qr()` die QR-Zerlegung berechnet und dann den OLS-Schätzer bestimmt, ohne die Funktion `lm()` zu nutzen. Testen Sie Ihren Code an einem beliebigen Datensatz und vergleichen Sie das Ergebnis mit dem der Funktion `lm()`.

Hinweis: Die Funktion soll eine Matrix von erklärenden und einen Vektor der zu erklärenden Variable als Argument übergeben bekommen. Trennen Sie den Datensatz bevor Sie die Funktion anwenden dementsprechend auf. Sie können einen `data.frame` mit `as.matrix()` in eine Matrix umwandeln. Mit der Funktion `class()` können Sie sich die Struktur anzeigen lassen. Achten Sie darauf das der Vektor der zu erklärenden Variable als Klasse `matrix` oder `numeric` hat!
Hinweis: Sobald Sie `qr()` auf eine Matrix angewendet haben, können Sie mit `qr.Q()` und `qr.R()` auf die Matrizen Q und R zugreifen: `qr.Q(qr(X))`.

Aufgabe 4

Eine der Hauptanwendungen für Regressionsanalysen ist die Durchführung von Prognosen. Wir unterscheiden 2 Arten von Prognosen:

- Prognosen für die Beobachtungen die genutzt wurden, um das Modell zu schätzen (**in-sample** Prognose).
- Prognosen für neue Beobachtungen, die bei der Anpassung des Modells nicht mit berücksichtigt wurden (**out-of-sample**).

Die **in-sample** Prognosen erhält man am einfachsten durch `fitted(lm_model)`. Um die abhängige Variable für neue Beobachtungen vorherzusagen (vorausgesetzt die Werte für die unabhängigen Variablen sind bekannt), kann die Funktion `predict()` genutzt werden. Betrachten Sie für diese Aufgabe folgenden R-Code:

```
set.seed(11)
data <- data.frame(
  x = runif(200, 0, 10)
)
data$y <- 4 + 3.5 * data$x + rnorm(200, 0, 20)
```

1. Auf Grundlage welches DGPs wurden die Daten erzeugt? Schätzen Sie anhand von 150 zufällig ausgewählten Beobachtungen die Parameter eines korrekt spezifizierten Modells. Nutzen Sie dann die Funktion `predict()`, um die übrigen 50 Beobachtungen vorherzusagen. Versuchen Sie selber herauszufinden, wie `predict()` funktioniert. Die Hilfeseite `?predict.lm` könnte hierbei hilfreich sein.
2. Berechnen Sie den MSE für die **in-sample** und die **out-of-sample** Prognosen.