Hinz, Adam Sommersemester 2025

Angewandte Statistik — Abgabezettel 1

(Ausgabe: 30.05.2025 — Abgabe: 06.06.2025)

Hinweise zur Abgabe dieses Aufgabenblattes

- Es sind nur **Einzelabgaben** zulässig (d.h. keine Gruppenarbeit).
- Ihre Lösung müssen Sie bis zum 06.06.2025 ("AoE, anywhere on earth") per **E-Mail** von Ihrer @uni-bielefeld.de-E-Mail-Adresse an abgabe-internationale-vwl@uni-bielefeld.de einreichen. Lösungen, die über andere E-Mail-Adressen eingereicht werden, werden nicht bewertet. Wählen Sie als Betreff "AngStat1 - 1234567", wobei Sie die Zahl "1234567" durch Ihre Matrikelnummer ersetzen.
- Ihre Lösung müssen Sie entweder als Quarto-Markdown-Dokument (**.qmd**) oder als R-Skript (**.R**) einreichen. Lösungen, die in anderen Dateiformaten eingereicht werden, werden nicht bewertet. Wählen Sie als Dateiname "AngStat1_1234567.qmd" bzw. "AngStat1_1234567.R", wobei Sie die Zahl "1234567" durch Ihre Matrikelnummer ersetzen. Es ist genau **eine** Datei einzureichen.
- Sie dürfen sich untereinander zu den Aufgaben austauschen, aber Achtung: nahezu wortgleiche Lösungen werden — sofern die Ähnlichkeit nicht plausibel erklärbar ist — als **Plagiat** gewertet, und zwar bei allen beteiligten Personen! (Also: sprechen Sie ruhig miteinander, aber passen Sie auf, dass niemand Ihre Lösungen kopiert.)
- Large Language Models (z.B. **ChatGPT**) dürfen Sie zur Unterstützung verwenden (z.B. bei Fehlermeldungen), nicht jedoch, um ganze Aufgaben zu lösen. Wenn nicht plausibel erklärbar ist, dass Sie Ihre Lösung selbständig erarbeitet haben, wird diese wie ein Plagiat gewertet.
- Dokumentieren Sie Ihre Schritte gut mittels Textblöcken in Ihrem Quarto-Markdown-Dokument oder Kommentaren in Ihrem R-Skript. Stellen Sie sicher, dass Ihr R-Code nicht nur auf Ihrem Computer, sondern auch bei uns durchläuft (starten Sie dazu vor der Einreichung RStudio neu und lassen Sie Ihre Lösung noch einmal durchlaufen). Für nicht **reproduzierbare** Lösungen gibt es 0 Punkte.
- Zur **Bewertung**: bei jeder Teilaufgabe ist 1 Punkt zu vergeben — einzelne kleinere Fehler bei grundsätzlich richtigen Lösungswegen führen dazu, dass 0.5 Punkte vergeben werden. Bei größeren Fehlern oder mehreren kleineren Fehlern innerhalb einer Teilaufgabe gibt es 0 Punkte für die Teilaufgabe (auch dann, wenn Teile der Lösung richtig sind).

Aufgabe 1: Verteilungsschätzung (3 Punkte)

In dieser Aufgabe beschäftigen wir uns mit den täglichen Renditen von 2 verschiedenen Assets: ein Exchange Traded Fund (ETF) auf den MSCI World ("MSCI_World_Rendite") und Bitcoin ("Bitcoin_Rendite"). Nutzen Sie zunächst die folgenden Befehle, um Ihren Datensatz zu laden:

```
set.seed(1234567)
renditen = read.csv("https://tinyurl.com/5ta63m93")[sample(x = 1:1000, size = 250),]
```

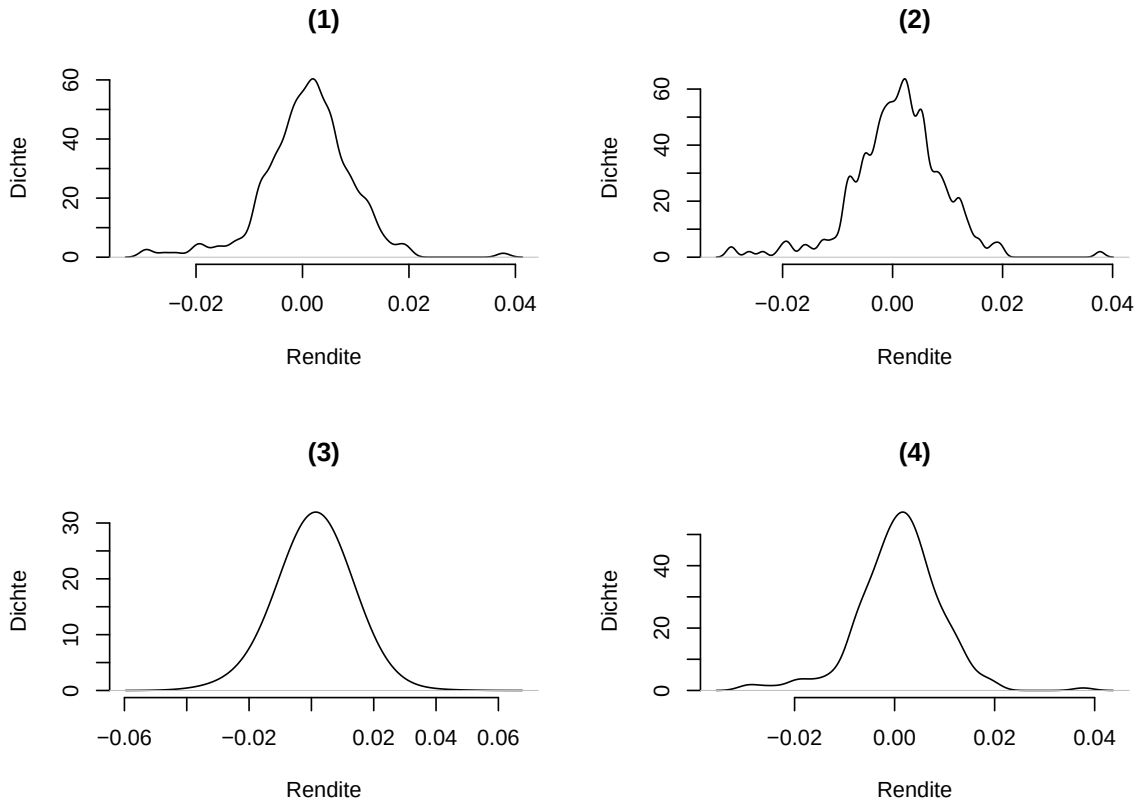
Ersetzen Sie hierbei die Zahl "1234567" durch Ihre Matrikelnummer. Ihr Datensatz enthält $n = 250$ Beobachtungen, wobei jede Beobachtung zu einem Tag gehört.

a) Bestimmen Sie für "MSCI_World_Rendite" und "Bitcoin_Rendite" die empirischen Verteilungsfunktionen. Vervollständigen Sie dann die folgenden Sätze:

- An 10% der Tage ist die Rendite des MSCI World-ETF _____ % oder höher.
- An _____ % der Tage ist die Rendite von Bitcoin -5% oder niedriger.
- An _____ % der Tage liegt die Rendite des MSCI World-ETF zwischen -1% und 1%.

b) Erstellen Sie für "MSCI_World_Rendite" und "Bitcoin_Rendite" Histogramme mit geeigneten Klassenbreiten, wobei auf den y-Achsen die Dichten abgetragen werden sollen. Fügen Sie dann Kerndichteschätzer zu den Histogrammen hinzu. Wählen Sie dafür eine geeignete Kernfunktion und geeignete Bandweiten und begründen Sie Ihre Wahl.

c) Schauen Sie sich die folgenden Kerndichteschätzer der MSCI World-ETF-Renditen an:



- In welcher Grafik ist der Bias am kleinsten? In welcher Grafik ist die Varianz am kleinsten?
- Erläutern Sie, warum Bias und Varianz nicht simultan minimiert werden können.

Aufgabe 2: Poisson-Regression (4 Punkte)

In dieser Aufgabe beschäftigen wir uns mit der täglichen Anzahl von Verkehrstoten in Spanien. Nutzen Sie zunächst die folgenden Befehle, um Ihren Datensatz einzulesen:

```
set.seed(1234567)
verkehr = read.csv("https://tinyurl.com/ycprh5cj")[sample(x = 1:365, size = 250),]
```

Ersetzen Sie hierbei wieder die Zahl "1234567" durch Ihre Matrikelnummer. Ihr Datensatz enthält $n = 250$ Beobachtungen, wobei jede Beobachtung zu einem Tag gehört. Die Zielvariable "Verkehrstote" gibt die Anzahl der Verkehrstoten an. Die erklärenden Variablen sind:

- "Temperatur": die Temperatur (in Grad Celsius).
- "Niederschlagsmenge": die Niederschlagsmenge (in ml/m^2).

a) Warum eignet sich ein klassiches lineares Modell nicht zur Modellierung der Zielvariable "Verkehrstote"? Passen Sie mithilfe der `glm()`-Funktion ein Poisson-Regressionsmodell mit der Zielvariable "Verkehrstote" und den erklärenden Variablen "Temperatur" und "Niederschlagsmenge" an die Daten an.

b) Nutzen Sie den `summary()`-Befehl, um die Koeffizienten des in a) geschätzten Modells auszugeben. Wie verändert sich die erwartete Anzahl der Verkehrstoten, wenn die Temperatur um 1 Grad Celsius steigt? Wie verändert sich die erwartete Anzahl der Verkehrstoten, wenn sich die Niederschlagsmenge um 1 ml/m^2 erhöht?

c) Wie hoch ist die erwartete Anzahl der Verkehrstoten an einem regenfreien Tag mit einer Temperatur von 20 Grad Celsius? Wie hoch ist die erwartete Anzahl der Verkehrstoten an einem Tag mit einer Temperatur von 20 Grad Celsius und einer Niederschlagsmenge von 200 ml/m^2?

d) Zeichnen Sie (z.B. mithilfe des `curve()`-Befehls) die Regressionsfunktion des in a) geschätzten Modells, wobei der Wert für "Niederschlagsmenge" auf 200 gesetzt werden soll.

Hinweis: in Ihrem Plot soll "E(Verkehrstote)" (also die erwartete Anzahl der Verkehrstoten) auf der y-Achse und "Temperatur" auf der x-Achse abgetragen werden.

Aufgabe 3: Logistische Regression (4 Punkte)

Ein Problem für E-Commerce-Unternehmen sind Betrugsversuche, bei denen bestellte Waren nicht bezahlt werden. In dieser Aufgabe beschäftigen wir uns mit Daten zu Bestellungen in einem Onlineshop. Nutzen Sie zunächst die folgenden Befehle, um Ihren Datensatz einzulesen:

```
set.seed(1234567)
ecommerce = read.csv("https://tinyurl.com/5n6at72u")[sample(x = 1:1000, size = 250),]
```

Ersetzen Sie hierbei wieder die Zahl "1234567" durch Ihre Matrikelnummer. Ihr Datensatz enthält $n = 250$ Beobachtungen, wobei jede Beobachtung zu einer Bestellung gehört. Die Zielvariable "Betrug" gibt an, ob es sich um einen Betrugsversuch (=1) oder um eine normale Bestellung (=0) gehandelt hat. Die erklärenden Variablen sind:

- "neuer_Kunde": Neukund*in ja (=1) oder nein (=0).
- "Rechnung": wurde auf Rechnung bestellt (=1) oder nicht (=0).
- "Menge": Anzahl an bestellten Teilen.
- "Betrag": Wert der Bestellung (in Euro).

a) Warum eignet sich ein Poisson-Regressionsmodell nicht zur Modellierung der Zielvariable "Betrug"? Passen Sie mithilfe der `glm()`-Funktion ein logistisches Regressionsmodell mit der Zielvariable "Betrug" und den erklärenden Variablen "neuer_Kunde", "Rechnung", "Menge" und "Betrag" an die Daten an.

b) Nutzen Sie den `summary()`-Befehl, um die Koeffizienten des in a) geschätzten Modells auszugeben. Interpretieren Sie die geschätzten Koeffizienten von "Menge" und "Betrag". Um wieviel Prozent erhöht sich die Betrugswahrscheinlichkeit für Neukund*innen?

c) Zeichnen Sie (z.B. mithilfe des `curve()`-Befehls) die Regressionsfunktion des in a) geschätzten Modells für Neukund*innen, die auf Rechnung bestellen. Der Wert für "Betrag" soll dabei auf 150 gesetzt werden.

Hinweis: in Ihrem Plot soll "P(Betrug)" (also die Wahrscheinlichkeit für einen Betrug) auf der y-Achse und "Menge" auf der x-Achse abgetragen werden.

d) Betrachten Sie einen Neukunden, der 3 Teile auf Rechnung bestellt. Bei welchem Rechnungsbetrag prognostiziert Ihr Modell aus a) eine Betrugswahrscheinlichkeit von 10%? Das Ergebnis sollen Sie anhand des geschätzten Modells mathematisch herleiten, eine Lösung durch Ausprobieren reicht nicht aus!

Hinweis: setzen Sie dafür die geschätzten Koeffizienten in die Modellgleichung ein, setzen Sie für die Betrugswahrscheinlichkeit 0.1 ein und lösen Sie die Modellgleichung nach "Betrag" auf.

Aufgabe 4: Nicht-parametrische Regression (4 Punkte)

In dieser Aufgabe beschätigen wir uns mit Daten zu Airbnb-Unterkünften in Amsterdam. Nutzen Sie zunächst die folgenden Befehle, um Ihren Datensatz einzulesen:

```
set.seed(1234567)
airbnb = read.csv("https://tinyurl.com/4fymh4ku")[sample(x = 1:500, size = 100),]
```

Ersetzen Sie hierbei wieder die Zahl "1234567" durch Ihre Matrikelnummer. Ihr Datensatz enthält $n = 100$ Beobachtungen, wobei jede Beobachtung zu einer Airbnb-Unterkunft gehört. Zu jeder dieser Unterkunft ist der "Preis" (in Euro) und die "Distanz" zum Stadtzentrum (in km) angegeben.

a) Nutzen Sie die `ksmooth()`-Funktion, um einen nicht-linearen Effekt von "Distanz" auf "Preis" mit einem Nadaraya-Watson-Schätzer (NWS) zu schätzen. Verwenden Sie dafür den Gaußkern. Wählen Sie per Augenmaß eine geeignete Bandweite und begründen Sie Ihre Wahl vor dem Hintergrund des Bias-Varianz-Tradeoffs.

b) Nutzen Sie die `gam()`-Funktion aus dem R-Paket `mgcv`, um einen nicht-linearen Effekt von "Distanz" auf "Preis" mithilfe von P-Splines zu schätzen. Plotten und interpretieren Sie den geschätzten Effekt.

c) Begründen Sie anhand des in b) geschätzten Modells, warum der Grad der für die Schätzung verwendeten Basisfunktionen nicht kleiner als 2 sein kann. Welche Konsequenzen hätten eine Erhöhung der Anzahl der Knoten/Basisfunktionen und eine Verringerung des Glättungsparameters?

d) Berechnen Sie für die angepassten Modelle aus a) und b) die Kreuzvalidierungskriterien

$$\text{CV(NWS)} = \sum_{i=1}^{n} \left(\text{Preis}_i - \hat{m}_{\text{NWS},-i}\left(\text{Distanz}_i\right)\right)^2$$

$$\text{CV(P-Spline)} = \sum_{i=1}^{n} \left(\text{Preis}_i - \hat{m}_{\text{P-Spline},-i}\left(\text{Distanz}_i\right)\right)^2,$$

wobei $\hat{m}_{\text{NWS},-i}\left(\text{Distanz}_i\right)$ und $\hat{m}_{\text{P-Spline},-i}\left(\text{Distanz}_i\right)$ der mithilfe eines NWS bzw. P-Splines prognostizierte Preis ist. Welches der beiden Modelle würden Sie, basierend auf Ihren Ergebnissen, bevorzugen?

Hinweis: wie man Prognosen aus einem NWS bekommt, haben Sie in der Vorlesung und im Tutorium bereits gesehen. Für P-Splines können Sie dafür die `predict()`-Funktion verwenden: `predict(mod, newdata = data.frame(Distanz = 1))`, wobei `mod` Ihr in b) geschätztes Modell ist und Sie für `1` beliebige Distanzen einsetzen können.