

Lehrstuhl für Ökonometrie
Prof. Dr. Christoph Hanck
M.Sc. Alexander Gerber

Klausur Zeitreihenanalyse

6. August 2018

Bitte bearbeiten Sie 3 der 4 Aufgaben. Falls Sie mehr als 3 Aufgaben bearbeiten, geben Sie bitte deutlich an, welche 3 Aufgaben bewertet werden sollen.

Die Bearbeitungszeit beträgt 60 Minuten (zzgl. 10 Minuten Einlesezeit). Sie können maximal 60 Punkte erreichen.

Führen Sie nur dann explizite Berechnungen durch wenn nötig. Nutzen Sie andernfalls allgemeine Eigenschaften.

Langwierige Antwortsätze, Abschreiben des Aufgabentextes etc. sind nicht nötig!

Wenn Sie keinen vollständigen Lösungsweg angeben können, skizzieren Sie bitte zumindest die Idee einer Lösung.

Die Nutzung des Internets ist nicht gestattet!

Viel Erfolg!

1 [20 Punkte]

```
(a) manager_rendite <- colMeans(Rendite)
sum(manager_rendite > 0)

## [1] 60
```

60 Manager haben über den betrachteten Zeitraum eine positive Überschussrendite erzielt. Das kann jedoch auf reines Glück zurückzuführen sein und muss nicht heißen, dass diese Manager auch in Zukunft den Markt schlagen werden.

(b) Mit einem t-test kann überprüft werden, welche der Manager eine statistisch signifikante Überschussrendite erzielt haben.

```
t_stat <- function(y, mu_0){
  sqrt(length(y)) * (mean(y) - mu_0) / sd(y)
}

tstat <- NULL
pval <- NULL
for(i in 1:ncol(Rendite)){
  tstat[i] <- t_stat(Rendite[,i], mu_0 = 0)
  pval[i] <- 1 - pnorm(tstat[i])
}
sum(pval < 0.05)

## [1] 7
```

```
## Alternativ kann auch die Funktion t.test benutzt werden
pval2 <- NULL
for(i in 1:ncol(Rendite)){
  pval2[i] <- t.test(Rendite[,i], mu = 0, alternative = "greater")$p.value
}
sum(pval2 < 0.05)

## [1] 7
```

Insgesamt haben 7 Manager eine signifikant positive Überschussrendite erzielt.

- (c) Basierend auf diesen Tests wäre es trotzdem nicht ratsam einem der Manager sein Geld anzuvertrauen, da ein Problem durch multiples Testen besteht. Je mehr Tests man durchführt, umso wahrscheinlicher wird es, dass eine Hypothese fälschlicherweise abgelehnt wird. Im Fall, dass die Nullhypothese zutrifft, wird diese mit einer Wahrscheinlichkeit von $\alpha * 100\%$ verworfen. Beim durchführen vieler Tests, steigt die Gefahr einen Fehler 1. Art zu machen (Alphafehler-Kumulierung).
- (d) Die in b) durchgeführten Tests sind unter Autokorrelation nicht robust was zu falschen Testergebnissen führen kann. Unter diesen Umständen sollte ein HAC-Tests benutzt werden. Anstelle der Standardabweichung wird hier die Wurzel der geschätzten Long-Run Variance verwendet.

```
l_T <- floor(4*(nrow(Rendite)/100)^(2/9))

#Bartlett Kernel von Folie 3-35
k <- function(j,l) {
  if(j <= l - 1) { return(1 - j/l) }
  if(j > l - 1) { return(0) }
}

gamma_hat <- acf(Rendite[, ], type = "covariance", lag.max = l_T, plot = FALSE)$acf
J <- gamma_hat[1] + 2*(gamma_hat[2] * k(1,l_T) +
  gamma_hat[3] * k(2,l_T) +
  gamma_hat[4] * k(3,l_T))
#gamma_hat[1] <=> gamma_0, gamma_hat[2] <=> gamma_1, usw.
t_HAC <- sqrt(length(Rendite[,1])) * (mean(Rendite[,1]) - 0) / sqrt(J)
1 - pnorm(t_HAC)

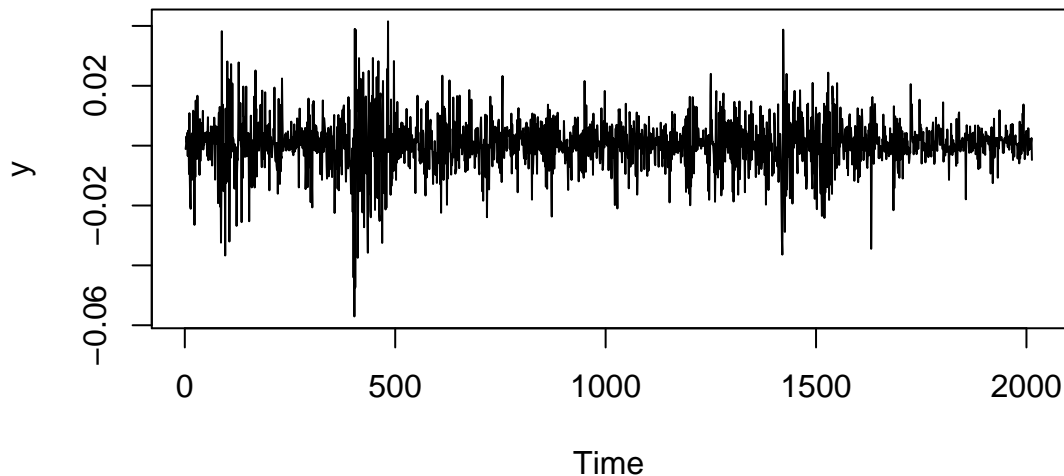
## [1] 0.2912871
```

2 [20 Punkte]

```
library(fGarch)
load("Dow.rda")
```

- (a) Die Zeitreihe sieht nicht stationär aus. Da es sich um einen Aktienindex handelt, liegt wahrscheinlich ein stochastischer Trend vor. Nur anhand der Grafik könnte man jedoch auch einen deterministischen Aufwärtstrend vermuten.

```
# Berechnung der log-Differenzen
y <- diff(log(Dow))
plot(y)
```



Anders als der Kurs sehen die log-Differenzen stationär aus. Zudem gibt es Volatilitätscluster.

- (b) Die Klasse der GARCH-Modelle eignet sich, um Zeitreihen mit Volatilitätsclustern zu modellieren. Der folgende Code passt ein ARCH(1) und ein GARCH(1, 1) an die Daten an.

```
garch10.fit <- garchFit(formula = ~ garch(1, 0), data = y, trace = FALSE)
garch11.fit <- garchFit(formula = ~ garch(1, 1), data = y, trace = FALSE)
```

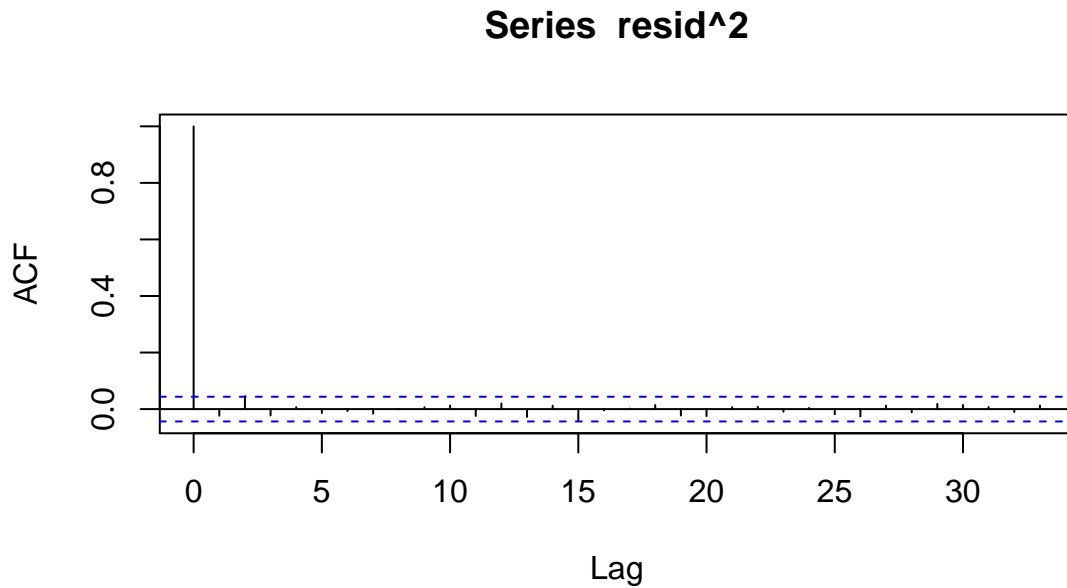
Über `summary()` erhält man für jedes Modell das AIC.

```
summary(garch10.fit)
summary(garch11.fit)
```

Für das ARCH(1) Modell erhalten wir ein AIC von -6.725613 und für das GARCH(1, 1) von -6.952842. Wir entscheiden uns für das GARCH(1, 1), da es das kleinere AIC hat.

- (c) Um zu überprüfen wie gut der Fit unseres Modells ist, schauen wir uns die standardisierten Residuen ($\hat{\epsilon}_t = y_t / \hat{\sigma}_t$) genauer an (vgl. Folie 5-28). Wenn $\hat{\epsilon}_t$ nach i.i.d. aussieht, ist unser Modell akzeptabel. Wir können hier z.B. einen Blick auf die ACF von $\hat{\epsilon}_t$ und $\hat{\epsilon}_t^2$ werfen oder einen Ljung-Box Test durchführen (`summary()` #Standardised Residuals Tests)

```
resid <- garch11.fit$residuals / garch11.fit$sigma.t
acf(resid^2)
```



Wir sehen, dass von über 30 berechneten Autokorrelationen nur eine zum 5%-Niveau signifikant ist. Demzufolge würden wir zu dem Schluss kommen, dass unsere Residuen White Noise sind. Außerdem wird die H_0 : „Die Residuen sind White Noise“ von allen durchgeführten Ljung-Box Tests abgelehnt.

- (d) Der $CVaR_\alpha$ ist der Verlust, der basierend auf den vorhandenen Informationen nur mit einer Wahrscheinlichkeit von $\alpha \cdot 100\%$ überschritten wird. Mit einer Vorhersage für σ_{T+1} kann $CVaR_\alpha$ durch

$$\widehat{CVaR}_\alpha = \hat{\sigma}_{T+1} \cdot \Phi^{-1}(\alpha)$$

geschätzt werden. In R können wir basierend auf unserem bereits angepassten Modell $\hat{\sigma}_{T+1}$ mithilfe von `predict()` berechnen.

```
sigma <- predict(garch11.fit, 1)$standardDeviation
CVaR   <- sigma * qnorm(0.05)
CVaR
## [1] -0.008101326
```

3 [20 Punkte]

- (a) Mit der Kettenregel erhält man:

$$\begin{aligned}\hat{y}_{t+1} &= \phi_1 y_t + \phi_2 y_{t-1} \\ \hat{y}_{t+2} &= \phi_1 \hat{y}_{t+1} + \phi_2 y_t = (\phi_1^2 + \phi_2) y_t + \phi_1 \phi_2 y_{t-1}\end{aligned}$$

Bei einer optimalen Vorhersage, ist der Vorhersagefehler (e_{t+s}) nicht mit den Informationen die zum Zeitpunkt t verfügbar sind korreliert.

$$E[\mathbf{X}_t e_{t+s}] = \mathbf{0}$$

- (b) Unter der Vorhersage aus der Aufgabenstellung ist der Vorhersagefehler

$$e_{t+1} = \epsilon_{t+1} + \phi_2 y_{t-1}.$$

Bis zum Zeitpunkt t stehen folgende Informationen zur Verfügung

$$\mathbf{X}_t = [y_t, y_{t-1}, \dots]^\top.$$

$$\begin{pmatrix} E[y_t e_{t+1}] \\ E[y_{t-1} e_{t+1}] \end{pmatrix} = \begin{pmatrix} E[y_t(\epsilon_{t+1} + \phi_2 y_{t-1})] \\ E[y_{t-1}(\epsilon_{t+1} + \phi_2 y_{t-1})] \end{pmatrix} = \begin{pmatrix} \phi_2 E[y_t y_{t-1}] \\ \phi_2 E[y_{t-1}^2] \end{pmatrix} = \begin{pmatrix} \phi_2 \gamma_0 \\ \phi_2 \gamma_1 \end{pmatrix} \neq \mathbf{0}$$

Offensichtlich sind e_{t+1} und \mathbf{X}_t korreliert. Damit kann die Vorhersage nicht optimal sein.

- (c) Das Vorhersageintervall kann wie folgt berechnet werden:

$$\hat{Y}_{t+s} \pm \phi^{-1} \left(1 - \frac{\alpha}{2}\right) \sqrt{MSE(s)}$$

wobei $MSE(s) = E[e_{t+s}^2]$. Der Vorhersagefehler ist $e_{t+s} = y_{t+s} - \hat{y}_{t+s}$. Damit erhält man

$$\begin{aligned} e_{t+1} &= \epsilon_{t+1} \\ e_{t+2} &= \phi_1 \epsilon_{t+1} + \epsilon_{t+2} \end{aligned}$$

woraus folgt, dass

$$\begin{aligned} MSE(1) &= \sigma^2 \\ MSE(2) &= \sigma^2(1 + \phi_1^2) \end{aligned}$$

Durch einsetzen in die obige Gleichung erhält man das Vorhersageintervall.

- (d) In der praktischen Anwendung müssen die Modellparameter z.B. mit OLS geschätzt werden. Selbst wenn man das genaue Modell kennt (z.B. AR(2)) wird die Schätzung von den wahren Parameterwerten abweichen. Hierdurch wird die Varianz des Vorhersagefehlers größer. Die Intervalle die mit der Formel aus Aufgabenteil c) berechnet werden (wenn einfach ϕ_i durch $\hat{\phi}_i$ ersetzt wird) sind dann zu schmal und müssen korrigiert werden.

4 [20 Punkte]

- (a) Die m -te partielle Autokorrelation ($\alpha_m^{(m)}$) ist der letzte Koeffizient der linearen Projektion von $(Y_t - \mu)$ auf die m direkten Vorgänger

$$(\hat{Y}_t - \mu) = \alpha_1^{(m)}(Y_{t-1} - \mu) + \alpha_2^{(m)}(Y_{t-2} - \mu) + \dots + \alpha_m^{(m)}(Y_{t-m} - \mu).$$

Um die ersten n partiellen Autokorrelationen zu schätzen, müssen n Regressionsgleichungen für $m = 1, \dots, n$ aufgestellt werden. Die Koeffizienten können dann mit OLS geschätzt werden, wobei jeweils der m -te geschätzte Koeffizient der geschätzten m -ten partiellen Autokorrelation entspricht.

- (b)
- Prozess A \rightarrow ACF/PACF 5
 - Prozess B \rightarrow ACF/PACF 2
 - Prozess C \rightarrow ACF/PACF 4
 - Prozess D \rightarrow ACF/PACF 1
 - Prozess E \rightarrow ACF/PACF 3
- (c) Ein Beispiel für den ersten Prozess wäre

$$y_t = y_{t-1} = X, \quad X \sim N(0, 1).$$

Die zweite Autokovarianzfunktion gehört zu einem MA(1)-Prozess, was daran zu erkennen ist, dass alle Autokovarianzen nach dem 1-ten Lag gleich 0 sind. Die allgemeine Formel für die Autokovarianzfunktion

$$\text{eines MA(1) ist gegeben durch } \gamma(j) = \begin{cases} \sigma^2(1 + \theta^2), & j = 0 \\ \theta\sigma^2 & , j = \pm 1, \\ 0, & \text{sonst} \end{cases} \quad (\text{vgl. Skript Folie 2-1 und 2-2}).$$

Durch Gleichsetzen und Auflösen nach den Parametern erhält man zwei mögliche Prozesse:

$$y_t = 2\epsilon_{t-1} + \epsilon_t, \quad \epsilon_t \sim (0, 0.2)$$

$$y_t = 0.5\epsilon_{t-1} + \epsilon_t, \quad \epsilon_t \sim (0, 0.8)$$

(d) Für einen ergodischen Prozess gilt, dass

$$\bar{y} = \frac{1}{T} \sum_{t=1}^T Y_t \xrightarrow{P} E[Y_t].$$

Der erste Prozess ist nicht ergodisch, da $\bar{y} = X \xrightarrow{P} X \neq E[X]$. Der zweite Prozess ist ergodisch, da $\sum_{j=0}^{\infty} |\gamma_j| = 1,4 < \infty$ (vgl. Theorem 1.10).