SPRAWOZDANIE 3

EKONOMETRIA

MARTA SOMMER – BSMAD

Będziemy rozważać 16 portfeli i na tej podstawie modelować polską giełdę. Mamy do dyspozycji dwa zestawy danych posortowane odpowiednio względem różnych wskaźników. Będziemy chcieli sprawdzić, który model: jednoczynnikowy, trójczynnikowy, czy czteroczynnikowy, dobrze opisuje polską giełdę.

Model jednoczynnikowy

Rozważmy model jednoczynnikowy:

$$R_{i,t} = \alpha_i + \beta_{RM,i} RM_t + \varepsilon_{i,t},$$

gdzie $i = 1, \ldots, 16$, $t = 1, \ldots, 97$.

Dopasowaliśmy więc w ten sposób 16 portfeli dla pierwszego zestawu danych.

Heteroskedastyczność badaliśmy testem Breuscha-Pagana i ten nie wykazał heteroskedastyczności w żadnym modelu, natomiast test White'a wykrył heteroskedastyczność tylko w portfelu 16. Śmiało możemy więc stwierdzić, że nie mamy tu do czynienia z heteroskedastycznością. Co do autokorelacji, to, przy pomocy testu Durbina-Watsona, została ona wykryta w ponad połowie przypadków, czyli niestety mamy z nią do czynienia w modelu. Zatem w portfelach, w których autokorelacja została wykryta będziemy przy szacowaniu wariancji korzystać z poprawki Newey-Westa.

Zobaczmy teraz w jaki sposób układają się wartości estymatorów β_{RM}:

β_{RM}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV} \downarrow$	0.9553296	0.8540825	0.8334983	0.8029792
	1.008922	0.8993477	0.8797741	1.045466
	1.1172	1.058734	1.002255	1.137929
	1.09756	1.1529	1.152157	1.124232

Z powyższej tabeli nie widać wiele zależności. Jedynie to, że mniejsze spółki z większym $\frac{BV}{MV}$ (duża wartość księgowa i mała wartość rynkowa) są bardziej agresywne (a tym samym bardziej ryzykowne) niż spółki z małym współczynnikiem $\frac{BV}{MV}$.

Przejdźmy teraz do testu restrykcji GRS o hipotezie:

$$\begin{cases} H : \alpha = 0 \\ K : \alpha \neq 0 \end{cases}$$

$P - value$ tego testu jest małe i wynosi $0,016$, zatem odrzucamy hipotezę zerową, czyli $\alpha \neq 0$. Oznacza to tyle, że nasz model jednoczynnikowy źle opisuje polską giełdę, gdyż istnieje w nim element losowości, przypadkowości. Nie ma więc już sensu badać zestawu drugiego danych ani oszacowywać premii za ryzyko w modelu.

Model trójczynnikowy

Rozważmy model trójczynnikowy:

$$R_{i,t} = \alpha_i + \beta_{RM,i}RM_t + \beta_{SMB,i}SMB_t + \beta_{HML,i}HML_t + \varepsilon_{i,t},$$

gdzie $i = 1, \ldots, 16$, $t = 1, \ldots, 97$.

Dopasowaliśmy więc w ten sposób 16 portfeli dla pierwszego zestawu danych.

Heteroskedastyczność badaliśmy testem Breuscha-Pagana i ten wykazał heteroskedastyczność w trzech portfelach, natomiast test White'a wykrył heteroskedastyczność w pięciu portfelach. Należy więc stwierdzić, że heteroskedastyczność jest obecna. Co do autokorelacji, to, przy pomocy testu Durbina-Watsona, została ona wykryta tylko w dwóch przypadkach, czyli problem autokorelacji w modelu możemy pominąć. Mimo wszystko, ze względu na heteroskedastyczność, trzeba będzie skorzystać z poprawki Newey-Westa.

Zobaczmy teraz w jaki sposób układają się wartości estymatorów β:

β_{RM}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV} \downarrow$	0.9385604	0.8875909	0.8255328	0.8495374
	0.8592244	0.8457644	0.821443	1.036485
	1.052344	0.9245519	0.8528089	1.08829
	0.8839021	0.8837371	0.9143578	0.9778986

β_{SMB}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV} \downarrow$	1.348105	1.107419	0.3920486	0.0130365
	1.481069	0.9689015	0.2761607	0.2071676
	1.322681	0.8095409	0.1186516	−0.1282186
	1.176667	0.7678757	0.06623415	−0.4334654

Widać, że im większa spółka tym mniejsze ryzyko związane z SMB.

β_{HML}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV} \downarrow$	-0.322925	-0.4655263	-0.0808298	-0.2009929
	0.2011988	-0.05619854	0.1663436	-0.02250085
	-0.1118389	0.3317198	0.5982598	0.247697
	0.5610356	0.9155656	0.9877718	0.7464178

Najbardziej ryzykowne są spółki z dużym $\frac{BV}{MV}$ (widać tu taką zależność monotoniczną).

Przejdźmy teraz do testu restrykcji GRS o hipotezie:

$$\begin{cases} H : \alpha = 0 \\ K : \alpha \neq 0 \end{cases}$$

$P - value$ tego testu jest duże i wynosi 0.051, zatem nie mamy podstaw do odrzucenia hipotezy zerowej, czyli $\alpha = 0$. Oznaczałoby to tyle, że nasz model trójczynnikowy dobrze opisuje polską giełdę, gdyż nie istnieje w nim element losowości, przypadkowości. Sprawdźmy jeszcze jednak, czy dobrze opisuje on też drugi zestaw danych. Tym razem $p - value$ testu GRS jest już małe i wynosi 0.004, tak więc odrzucamy hipotezę. Model zatem nie sprawdza się dla drugiego zestawu danych. Model ten więc źle opisuje polski rynek.

Przejdźmy zatem do modelu czteroczynnikowego.

Model czteroczynnikowy

Rozważmy model trójczynnikowy:

$$R_{i,t} = \alpha_i + \beta_{RM,i} RM_t + \beta_{SMB,i} SMB_t + \beta_{HML,i} HML_t + \beta_{WML,i} WML_t + \varepsilon_{i,t},$$

gdzie $i = 1, \ldots, 16$, $t = 1, \ldots, 97$.

Dopasowaliśmy więc w ten sposób 16 portfeli dla pierwszego zestawu danych.

Heteroskedastyczność badaliśmy testem Breuscha-Pagana i ten wykazał heteroskedastyczność w dwóch portfelach, natomiast test White'a wykrył heteroskedastyczność w pięciu portfelach. Należy więc stwierdzić, że heteroskedastyczność jest obecna. Co do autokorelacji, to, przy pomocy testu Durbina-Watsona, została ona wykryta tylko w dwóch przypadkach, czyli problem autokorelacji w modelu możemy pominąć. Mimo wszystko, ze względu na heteroskedastyczność, trzeba będzie skorzystać z poprawki Newey-Westa.

Zobaczmy teraz w jaki sposób układają się wartości estymatorów β:

β_{RM}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV}\downarrow$	0.9232306	0.8641042	0.8241291	0.8427869
	0.8479656	0.8568631	0.8152669	1.026009
	1.061216	0.9249988	0.8416023	1.092635
	0.8677172	0.8730714	0.9079475	0.9748165

Przy β_{RM} nie widać zbytniej zależności.

β_{SMB}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV}\downarrow$	1.343542	1.100427	0.3916307	0.011027
	1.477718	0.9722053	0.2743222	0.2040493
	1.325322	0.809674	0.1153156	−0.1269254
	1.171849	0.7647007	0.06432592	−0.4343829

β_{RM} układa się monotonicznie zarówno ze względu na kapitalizację, jak i ze względu na $\frac{BV}{MV}$. Przy czym najmniej ryzykowne są duże spółki z dużym $\frac{BV}{MV}$, a najbardziej ryzykowne małe spółki z małym $\frac{BV}{MV}$ (rynek trochę je przecenia, dlatego są ryzykowne).

β_{HML}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV}\downarrow$	−0.4544298	−0.6670027	−0.09287146	−0.2589012
	0.1046171	0.03900907	0.1133629	−0.1123617
	−0.03573682	0.3355533	0.502126	0.2849643
	0.4221964	0.8240721	0.932782	0.7199789

β_{RM} układa się w miarę monotonicznie ze względu na $\frac{BV}{MV}$. I tym razem najbardziej ryzykowne są duże spółki z dużym $\frac{BV}{MV}$.

β_{WML}	kapitalizacja $\longrightarrow$			
$\frac{BV}{MV}\downarrow$	−0.1596162	−0.2445455	−0.01461578	−0.07028727
	−0.1172278	0.1155599	−0.06430635	−0.1090702
	0.09237024	0.004652982	−0.1166842	0.04523386
	−0.1685186	−0.1110518	−0.06674487	−0.03209072

Przejdźmy teraz do testu restrykcji GRS o hipotezie:

$$\begin{cases} H: \alpha = 0 \\ K: \alpha \neq 0 \end{cases}$$

$P-value$ tego testu jest duże i wynosi 0.19, zatem nie mamy podstaw do odrzucenia hipotezy zerowej. Model więc dobrze opisuje polską giełdę. Sprawdźmy jednak, czy dobrze zachowuje się również w przypadku drugiego zestawu danych, bo być może, tak jak w przypadku modelu trójczynnikowego, nie będzie się dobrze zachowywał na nowych danych. $P-value$ wynosi 0.23. Jest więc duże i znów nie daje podstaw do odrzucenia hipotezy zerowej. Model czteroczynnikowy sprawdza się więc w przypadku polskiej giełdy.

Spróbujmy więc oszacować premię za ryzyko w tym modelu.

$$\mathbb{E}(R_t) = \gamma_0 + \gamma_{RM}\beta_{RM} + \gamma_{SMB}\beta_{SMB} + \gamma_{HML}\beta_{HML} + \gamma_{WML}\beta_{WML}$$

Metodą ważonych najmniejszych kwadratów otrzymujemy następujące wyniki:

$$\begin{aligned}
\gamma_0 &= -0.0327659584 \\
\gamma_{RM} &= 0.0273126639 \\
\gamma_{SMB} &= 0.0017693357 \\
\gamma_{HML} &= 0.0066998285 \\
\gamma_{WML} &= -0.0005393353
\end{aligned}$$

Przeprowadzając test istotności współczynników wyszło, że tylko γ_{RM} jest niezerowy. Czyli tylko on, gdy się zwiększy będzie miał wpływ na premię za ryzyko.

```r
library("bstats")
library("lmtest")
library("FinTS")

# model jednoczynnikowy:

p1 <- read.csv2("C:\\Users\\Marta\\Desktop\\Marta\\studia\\rok4\\Ekonometria\\spr3\\portfele_z1.csv",
    sep = ";", header = TRUE)
head(p1)

wig1 <- p1[, 2]

l1 <- vector("list", 16)
```

```
for (i in 1:16) {
    x <- p1[, i + 5]
    l1[[i]] <- lm(x ~ wig1)
}
l1

w1 <- numeric(16)
bg <- numeric(16)
gq <- numeric(16)
dw <- numeric(16)
for (i in 1:16) {
    w1[i] <- ifelse(white.test(l1[[i]])$p.value < 0.05, 1, 0)
    bg[i] <- ifelse(bptest(l1[[i]])$p.value < 0.05, 1, 0)
    gq[i] <- ifelse(gqtest(l1[[i]], fraction = 0.33, order.by = ~wig1)$p.value <
        0.05, 1, 0)
    dw[i] <- ifelse(dwtest(l1[[i]], alternative = "two.sided")$p.value < 0.05,
        1, 0)
}

w1  # heteroskedastycznosc
bg  # heteroskedastycznosc
gq  # heteroskedastycznosc
dw  # autokorelacja



rm1 <- numeric(0)
for (i in 1:16) {
    rm1[i] <- l1[[i]]$coefficients[2]
}

# tabele dla RM:

rm1
tabela1_rm <- matrix(rm1, nrow = 4)
tabela1_rm

t <- nrow(p1)
```

```
n <- 16
k <- 1

alfa1 <- numeric(16)
for (i in 1:16) {
    alfa1[i] <- l1[[i]]$coefficients[1]
}
alfa1

m <- matrix(0, nrow = 97, ncol = 16)
for (i in 1:16) {
    a <- l1[[i]]$residuals
    for (j in 1:97) {
        m[j, i] <- a[j]
    }
}
m
dim(m)

sigma <- (t(m) %*% m)/t
sigma

head(p1)
v <- var(p1$WIG)

mi <- mean(p1$WIG)

grs <- (t/n) * ((t - n - k)/(t - k - 1)) * ((alfa1 %*% solve(sigma) %*% t(t(alfa1)))/(1 +
    mi %*% solve(v) %*% t(t(mi))))
grs
1 - pf(grs, n, t - n - k)  # małe -> odrzucamy hipoteze, czyli alfa nie sa zerami



# model trójczynnikowy:

l2 <- vector("list", 16)
for (i in 1:16) {
```

```r
    x <- p1[, i + 5]
    l2[[i]] <- lm(x ~ wig1 + p1$SMB + p1$HML)
}

w1 <- numeric(16)
bg <- numeric(16)
gq <- numeric(16)
dw <- numeric(16)
for (i in 1:16) {
    w1[i] <- ifelse(white.test(l2[[i]])$p.value < 0.05, 1, 0)
    bg[i] <- ifelse(bptest(l2[[i]])$p.value < 0.05, 1, 0)
    gq[i] <- ifelse(gqtest(l2[[i]], fraction = 0.33, order.by = ~wig1)$p.value <
        0.05, 1, 0)
    dw[i] <- ifelse(dwtest(l2[[i]], alternative = "two.sided")$p.value < 0.05,
        1, 0)
}
w1
bg
gq
dw

rm2 <- numeric(0)
smb2 <- numeric(0)
hml2 <- numeric(0)

for (i in 1:16) {
    rm2[i] <- l2[[i]]$coefficients[2]
    smb2[i] <- l2[[i]]$coefficients[3]
    hml2[i] <- l2[[i]]$coefficients[4]
}

tabela2_rm <- matrix(rm2, nrow = 4)
tabela2_smb <- matrix(smb2, nrow = 4)
tabela2_hml <- matrix(hml2, nrow = 4)

tabela2_rm
tabela2_smb
```

```r
tabela2_hml


t <- nrow(p1)
n <- 16
k <- 3

alfa2 <- numeric(16)
for (i in 1:16) {
    alfa2[i] <- l2[[i]]$coefficients[1]
}
alfa2

m <- matrix(0, nrow = 97, ncol = 16)
for (i in 1:16) {
    a <- l2[[i]]$residuals
    for (j in 1:97) {
        m[j, i] <- a[j]
    }
}
sigma2 <- (t(m) %*% m)/t

h <- matrix(c(p1$WIG, p1$SMB, p1$HML), nrow = 97)

v <- cov(h)
mi <- apply(h, 2, mean)
mi
grs <- (t/n) * ((t - n - k)/(t - k - 1)) * ((alfa2 %*% solve(sigma2) %*% t(t(alfa2)))/(1 +
    mi %*% solve(v) %*% t(mi)))
grs
1 - pf(grs, n, t - n - k)


# dla p2

l22 <- vector("list", 16)
for (i in 1:16) {
```

```r
    x <- p2[, i + 5]
    l22[[i]] <- lm(x ~ wig1 + p2$SMB + p2$HML)
}

w1 <- numeric(16)
bg <- numeric(16)
gq <- numeric(16)
dw <- numeric(16)
for (i in 1:16) {
    w1[i] <- ifelse(white.test(l22[[i]])$p.value < 0.05, 1, 0)
    bg[i] <- ifelse(bptest(l22[[i]])$p.value < 0.05, 1, 0)
    gq[i] <- ifelse(gqtest(l22[[i]], fraction = 0.33, order.by = ~wig1)$p.value <
        0.05, 1, 0)
    dw[i] <- ifelse(dwtest(l22[[i]], alternative = "two.sided")$p.value < 0.05,
        1, 0)
}
w1
bg
gq
dw

rm2 <- numeric(0)
smb2 <- numeric(0)
hml2 <- numeric(0)

for (i in 1:16) {
    rm2[i] <- l22[[i]]$coefficients[2]
    smb2[i] <- l22[[i]]$coefficients[3]
    hml2[i] <- l22[[i]]$coefficients[4]
}

tabela2_rm <- t(matrix(rm2, nrow = 4))
tabela2_smb <- t(matrix(smb2, nrow = 4))
tabela2_hml <- t(matrix(hml2, nrow = 4))

tabela2_rm
tabela2_smb
```

```
tabela2_hml



t <- nrow(p2)
n <- 16
k <- 3

alfa2 <- numeric(16)
for (i in 1:16) {
    alfa2[i] <- l22[[i]]$coefficients[1]
}
alfa2

m <- matrix(0, nrow = 97, ncol = 16)
for (i in 1:16) {
    a <- l22[[i]]$residuals
    for (j in 1:97) {
        m[j, i] <- a[j]
    }
}
sigma2 <- (t(m) %*% m)/t

h <- matrix(c(p1$WIG, p1$SMB, p1$HML), nrow = 97)

v <- cov(h)
mi <- apply(h, 2, mean)
mi
grs <- (t/n) * ((t - n - k)/(t - k - 1)) * ((alfa2 %*% solve(sigma2) %*% t(t(alfa2)))/(1 +
    mi %*% solve(v) %*% t(mi))))
grs
1 - pf(grs, n, t - n - k)  # odrzucamy hipotezę

############################################################################ model czteroczynnikowy:

l3 <- vector("list", 16)
for (i in 1:16) {
    x <- p1[, i + 5]
```

```
    l3[[i]] <- lm(x ~ wig1 + p1$SMB + p1$HML + p1$WML)
}

l3

w1 <- numeric(16)
bg <- numeric(16)
gq <- numeric(16)
dw <- numeric(16)
for (i in 1:16) {
    w1[i] <- ifelse(white.test(l3[[i]])$p.value < 0.05, 1, 0)
    bg[i] <- ifelse(bptest(l3[[i]])$p.value < 0.05, 1, 0)
    gq[i] <- ifelse(gqtest(l3[[i]], fraction = 0.33, order.by = ~wig1)$p.value <
        0.05, 1, 0)
    dw[i] <- ifelse(dwtest(l3[[i]], alternative = "two.sided")$p.value < 0.05,
        1, 0)
}

w1  # heteroskedastycznosc
bg  # heteroskedastycznosc
gq  # heteroskedastycznosc
dw  # autokorelacja


rm3 <- numeric(0)
smb3 <- numeric(0)
hml3 <- numeric(0)
wml3 <- numeric(0)

for (i in 1:16) {
    rm3[i] <- l3[[i]]$coefficients[2]
    smb3[i] <- l3[[i]]$coefficients[3]
    hml3[i] <- l3[[i]]$coefficients[4]
    wml3[i] <- l3[[i]]$coefficients[5]
}
```

```r
tabela3_rm <- matrix(rm3, nrow = 4)
tabela3_smb <- matrix(smb3, nrow = 4)
tabela3_hml <- matrix(hml3, nrow = 4)
tabela3_wml <- matrix(wml3, nrow = 4)



tabela3_rm
tabela3_smb
tabela3_hml
tabela3_wml

t <- nrow(p1)
n <- 16
k <- 4

alfa3 <- numeric(16)
for (i in 1:16) {
    alfa3[i] <- l3[[i]]$coefficients[1]
}
alfa3

m <- matrix(0, nrow = 97, ncol = 16)
for (i in 1:16) {
    a <- l3[[i]]$residuals
    for (j in 1:97) {
        m[j, i] <- a[j]
    }
}
sigma3 <- (t(m) %*% m)/t

h <- matrix(c(p1$WIG, p1$SMB, p1$HML, p1$WML), nrow = 97)

v <- cov(h)
mi <- apply(h, 2, mean)
mi
grs <- (t/n) * ((t - n - k)/(t - k - 1)) * ((alfa3 %*% solve(sigma3) %*% t(t(alfa3)))/(1 +
    mi %*% solve(v) %*% t(t(mi))))
```

```r
grs
1 - pf(grs, n, t - n - k)

# dla p2:

l33 <- vector("list", 16)
for (i in 1:16) {
    x <- p2[, i + 5]
    l33[[i]] <- lm(x ~ p2$WIG + p2$SMB + p2$HML + p2$WML)
}

t <- nrow(p2)
n <- 16
k <- 4

alfa3 <- numeric(16)
for (i in 1:16) {
    alfa3[i] <- l33[[i]]$coefficients[1]
}
alfa3

m <- matrix(0, nrow = 97, ncol = 16)
for (i in 1:16) {
    a <- l33[[i]]$residuals
    for (j in 1:97) {
        m[j, i] <- a[j]
    }
}
sigma3 <- (t(m) %*% m)/t

h <- matrix(c(p2$WIG, p2$SMB, p2$HML, p2$WML), nrow = 97)

v <- cov(h)
mi <- apply(h, 2, mean)
mi
grs <- (t/n) * ((t - n - k)/(t - k - 1)) * ((alfa3 %*% solve(sigma3) %*% t(t(alfa3)))/(1 +
    mi %*% solve(v) %*% t(t(mi))))
```

```r
grs
1 - pf(grs, n, t - n - k)

# 6 dwoch pierwszych nie ma sensu - tylko trzeci

# pierwsza metoda: (wazona metoda najmniejszych kwadratow)

l3

xx <- matrix(0, nrow = 16, ncol = 5)
xx[, 1] <- 1
for (i in 1:16) {
    xx[i, 2:5] <- l3[[i]]$coefficients[2:5]
}

xx
sigma3

p1

r_sr <- apply(p1[6:21], 2, mean)
r_sr

gamma <- solve(t(xx) %*% solve(sigma3) %*% xx) %*% t(xx) %*% solve(sigma3) %*%
    r_sr
gamma  # to jest premia za ryzyko -> srednio 0.2 procenta

lm(r_sr ~ xx + 0, weights = diag(sigma3))

va <- (1/97) * (solve(t(xx) %*% solve(sigma3) %*% xx) + rbind(0, cbind(0, v)))
va

gamma[1]/sqrt(va[1, 1])
1 - pt(gamma[1]/sqrt(va[1, 1]), 97 - 5)  # gammma 0 wyszlo rowne zero
qt(0.95, 97 - 5)

gamma[2]/sqrt(va[2, 2])
```

```r
1 - pt(gamma[2]/sqrt(va[2, 2]), 97 - 5)

gamma[3]/sqrt(va[3, 3])
1 - pt(gamma[3]/sqrt(va[3, 3]), 97 - 5)

gamma[4]/sqrt(va[4, 4])
1 - pt(gamma[4]/sqrt(va[4, 4]), 97 - 5)

gamma[5]/sqrt(va[5, 5])
1 - pt(gamma[5]/sqrt(va[5, 5]), 97 - 5)

# jesli hml zwiekszy sie o jeden procent, to ma to wplyw na wartosc naszego
# portfele, jesli sie zwieksza pozostale to juz raczej nie

# druga metoda:

eps <- numeric(5 * 16)
mac <- matrix(0, nrow = 80, ncol = 97)

for (j in 1:97) {

    for (i in 1:16) {
        eps[i] <- l3[[i]]$residuals[j]
    }

    eps[17:32] <- eps[1:16] * p1[j, 2]
    eps[33:48] <- eps[1:16] * p1[j, 3]
    eps[49:64] <- eps[1:16] * p1[j, 4]
    eps[65:80] <- eps[1:16] * p1[j, 5]

    mac[, j] <- eps
}

mac

gt <- apply(mac, 1, sum)
gt
```