Empirical Methods in Finance Homework 4: Solution

Prof. Lars A. Lochstoer

TAs: Chady Gemayel and Mahyar Kargar

February 21, 2019

Problem 1: Risk Management using Value-at-Risk and GARCH models

Consider the daily prices for one share in a Deutsche Bank currency fund provided to you in the spreadsheet "`Currency_fund_prices.xlsx`". The maximum VaR over a 20-day trading period for currency trading was recently set at $100 million.

1. Specify and estimate a parsimonious model for the conditional volatility of the daily log returns on this currency position. You can abstract from variation in the conditional mean of returns. Carefully explain why you chose this model. Provide some evidence that this model is a good fit for the data.

   Suggested Solutions:

   Estimates for 16 different models are provided below. The estimated models consist of the standard GARCH, exponential GARCH (E-GARCH), integrated GARCH (I-GARCH), and GJR-GARHCH models, each with Gaussian ($\mathcal{N}$) and Student's t distributed errors, and with orders of (1, 1) and (2, 2). Table 1 shows the information criteria for these models. Based on the results, I selected the E-GARCH(1,1) with Student's t-distributed errors. Figure 1 shows the time series of conditional volatility (blue) versus the demeaned absolute value of the log return (gray). The E-GARCH(1,1) with student's t-distributed errors does a good job in following the time series of the volatility and is able to capture some of the extreme values we see in the data. However, as seen in Figure 2, the standardized residuals are qualitatively consistent with the student's t-distribution

	Order (1, 1)		Order (2, 2)	
Model - Dist.	AIC	BIC	AIC	BIC
GARCH - $\mathcal{N}$	-7.16	-7.16	-7.15	-7.15
GARCH - t	-7.23	-7.22	-7.21	-7.21
E-GARCH - $\mathcal{N}$	-7.17	-7.17	-7.16	-7.15
E-GARCH - t	-7.23	-7.23	-7.22	-7.21
I-GARCH - $\mathcal{N}$	-7.16	-7.16	-7.15	-7.15
I-GARCH - t	-7.23	-7.23	-7.22	-7.21
GJR-GARCH - $\mathcal{N}$	-7.16	-7.16	-7.15	-7.14
GJR-GARCH - t	-7.23	-7.23	-7.21	-7.21

Table 1: Information Criterion for Model Specifications

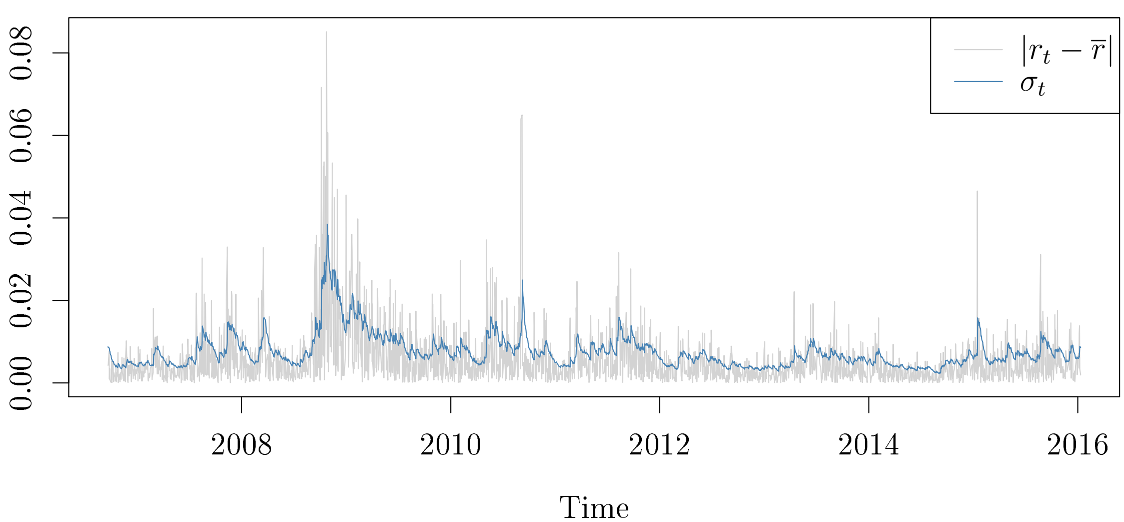

Figure 1: Conditional and Realized Volatility

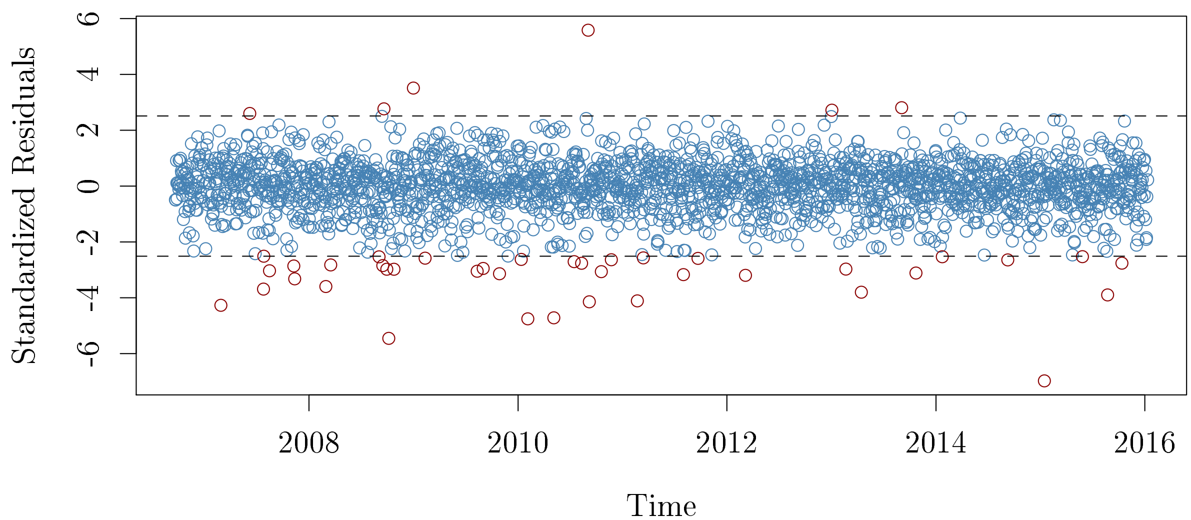

Figure 2: Standardized Residuals of the Fitted Model

2. Based on these estimates, develop a forecast for the 20-trading-day log return volatility on Jan 11, 2016 (end of day). Report the exact number and explain how you arrived at this number. You can assume daily returns (in levels) are independently distributed over time. (Hint: we assume the returns are uncorrelated across days. What then is the formula for 20-day log return volatility? It's the sum of each of the 20 individual days' expected variance.)

   Suggested Solutions: As we are dealing with log returns, the 20-trading-day log return will simply be the sum of the 20 predictions. And as we assume daily returns are independent, the variance of the 20-trading-day return will be the sum of each day's variance, such that

$$\mathbb{V}_t(r_{t,t+20}) = \sum_{i=1}^{20} \mathbb{V}_t(r_{t+i}) \tag{1}$$

   Figure 3 shows the predicted volatility. We can sum the predicted variances to get the variance of the 20-trading-day log return is 0.0012, meaning the volatility is 0.035.

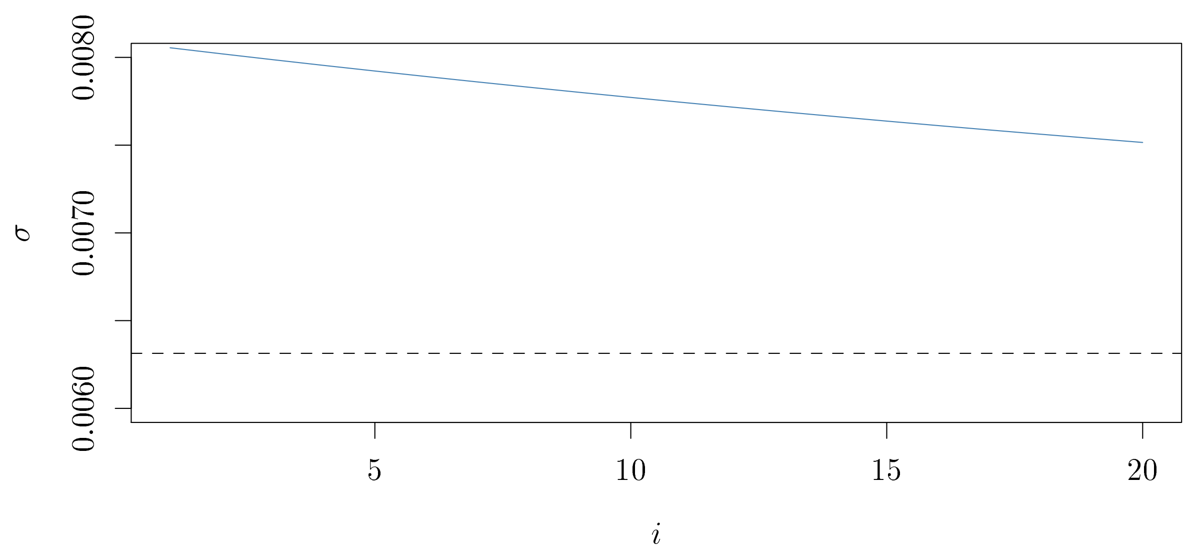

Figure 3: Conditional Volatility Forecast

Problem 2: The Single Factor (Market) Model

Download the 48 industry portfolio data (monthly) from Kenneth French's web site. Use the data from 1960 through 2015.[1] Use the value-weighted returns. You may drop the industries that have missing values and are reported as −99.99. Also, download the 3 Fama-French factors from his web site. Use the monthly risk-free rate series provided by French in the same FF factor dataset to compute excess returns on these 48 portfolios.

1. For each industry, regress the industry excess return on the market excess return (the FF market factor) and an intercept.

   (a) Plot the industry betas in a bar plot with industry number on the x-axis. For each bar, convey the ±2 standard error band of the beta. For instance, you can overlay lines using 'arrows' in R that give error bars (google "building barplots with error bars in R" for an example). The standard errors should be computed allowing for heteroskedastic and non-normal error terms.

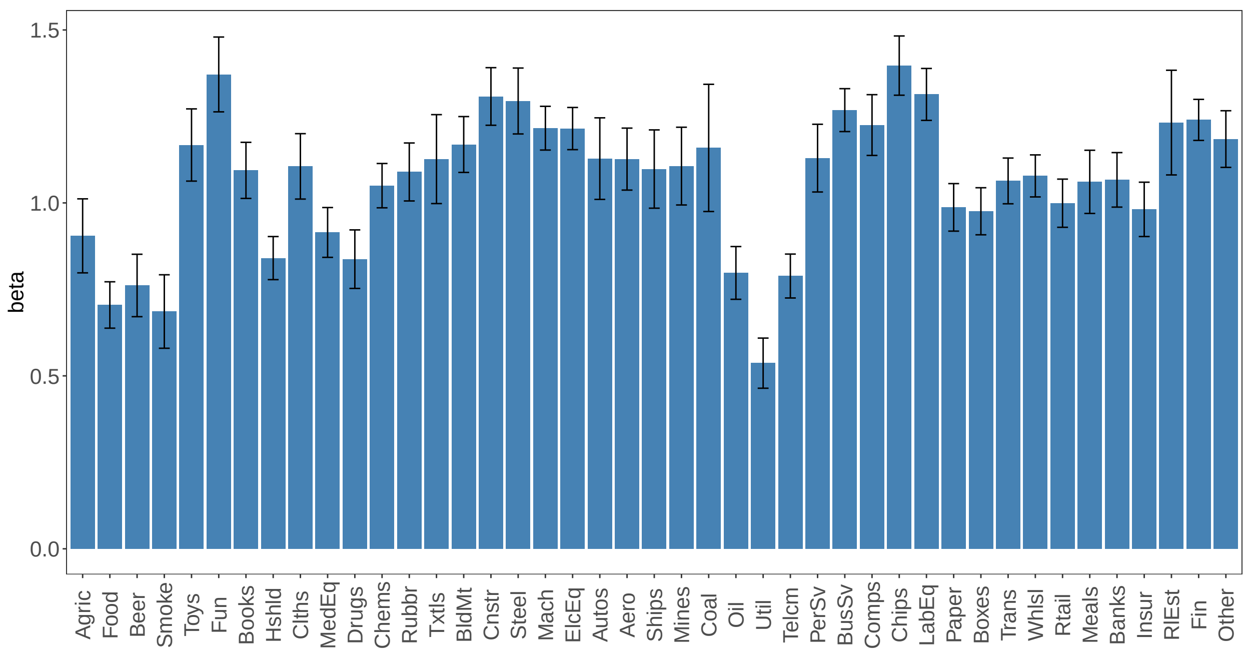

Figure 4: CAPM betas for different industries

Suggested solution:

[1] Of course, you can always try earlier periods to check the robustness of the results.

Figure 4 shows the beta for the one factor model (CAPM) across industries. The error bars are given as the ± 2 White standard error bands.

(b) What is the range of estimated betas? What are the min, max, and mean regression R^2 across industries?

Suggested solution:

Table 2 presents the minimum, maximum and mean for the estimated betas and R^2s across industries. *Chips* and *Util* have the highets and lowest estimated betas, while *Fin* and *Smoke* have the highets and lowest R^2s.

Table 2: Range of estimated betas and R^2s.

	β	R^2
min	0.537	0.249
	(Util)	(Smoke)
max	1.397	0.801
	(Chips)	(Fin)
mean	1.065	0.576

Note: Corresponding industries are in parentheses.

(c) Plot estimated alphas (intercept terms) against estimated betas. Is there a pattern? If so, can we guess at an example of a systemic failure of the CAPM?

Suggested solution:

Figure 5 shows the scatter plot of the estimated alphas and betas. First, it is not always true that beta is 1 and alpha is 0. Also, it seems the higher beta is, the lower alpha will be. This negative correlation between alphas and betas is evident from the negatively-sloped blue regression line indicating a systematic failure of the CAPM. Low beta stocks have too high returns and high beta stocks have too low returns relative to the CAPM. In fact, this a well-documnted anomaly in the finance literature and is called Betting Against Beta. See Frazzini and Pedersen (2014) for more details.[2]

2. Now, run rolling regressions of 5 years of data. That is, run first regressions using the data from 1960 through 1964, then from 1965 through 1969, etc. Note that the

[2] Frazzini, Andrea, and Lasse Heje Pedersen, 2014, Betting against beta, *Journal of Financial Economics* 111, 1-25.

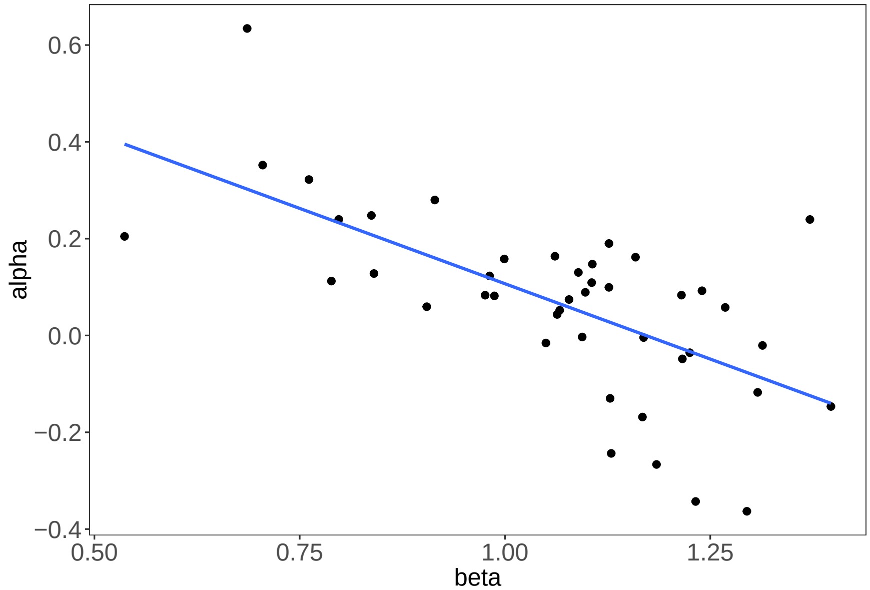

Figure 5: Scatter plot of CAPM alphas and betas.

last year, 2015, will not be used. You should now have 11 market betas per industry. Compute for each industry the correlations of adjacent betas (i.e., the beta from 1960-64 vs the beta from 1965-69; the beta from 1965-69 vs. the beta from 1970-74; etc.). Plot these correlations (y-axis) with industry number on the x-axis. Are the betas stable -i.e., highly correlated over time? What are the potential reasons they are not the same across 5-year periods?

Suggested solution:

The minimum, maximum, and mean correlations are -0.387 (for *LabEq*), 0.735 (for *Mach*), and 0.193, respectively. Figure 6 plots correlations of adjacent betas for different industries. Although some industries do show a stability over the sample period, most of the betas are *not* highly correlated with their adjacent ones. This low correlation could be attributed to time-varying betas.

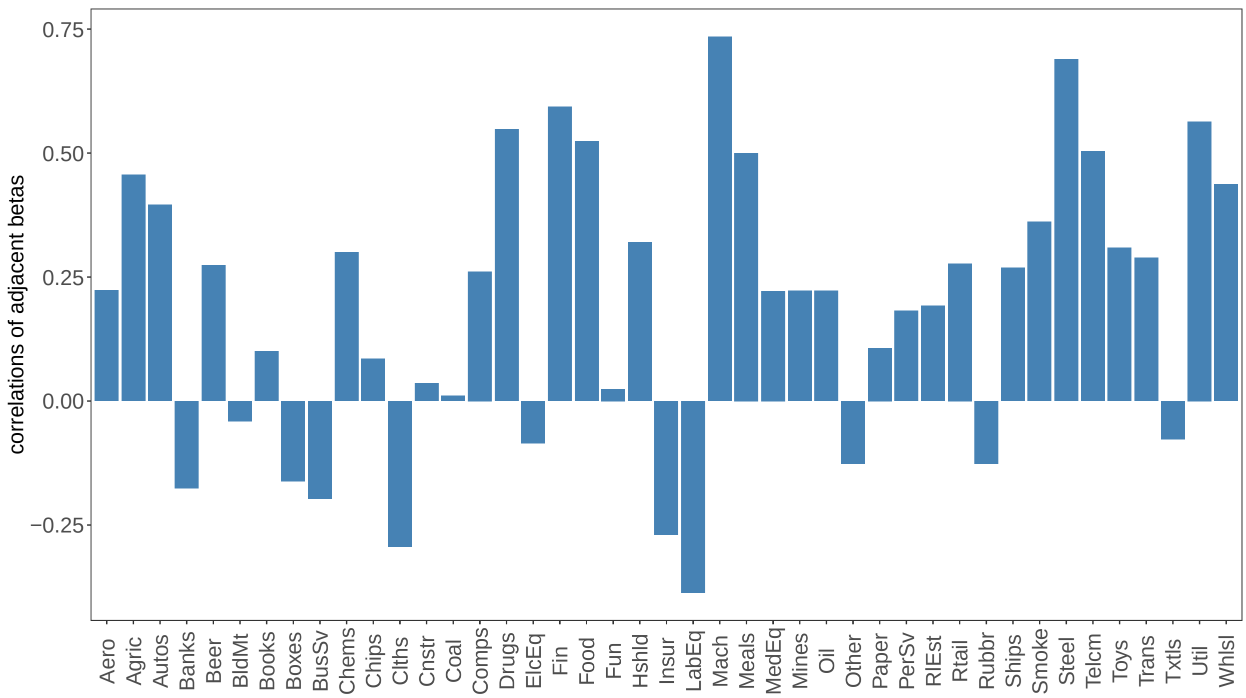

Figure 6: Correlation of adjacent betas

Suggested R Codes

```r
######################
# Code for Problem 1 #
######################

library(rugarch)
library(data.table)

currency <- fread("Currency_fund_prices.csv")
currency[, Date := as.Date(Date, "%m/%d/%y")]
currency[, lagP := shift(`Adj Close`, 1)]
currency[, logR := log(`Adj Close`/lagP)]

# use different GARCH models to fit the data

# GARCH(1,1) with norm errors
a.spec=ugarchspec(variance.model <- list(model="fGARCH",submodel="GARCH",
                                         garchOrder=c(1,1)),
                  mean.model = list(armaOrder=c(0,0)))

# GARCH(1,1) with student's t errors
b.spec <- ugarchspec(variance.model = list(model="fGARCH", submodel="GARCH",
                                           garchOrder=c(1,1)),
                     mean.model = list(armaOrder=c(0,0)),
                distribution.model = "std")

# EGARCH(1,1) with norm errors
c.spec <- ugarchspec(variance.model = list(model="eGARCH",
                                      garchOrder=c(1,1)),
                mean.model = list(armaOrder=c(0,0)))

# EGARCH(1,1) with student's t errors
d.spec <- ugarchspec(variance.model = list(model="eGARCH", garchOrder=c(1,1)),
                     mean.model = list(armaOrder=c(0,0)),
                  distribution.model = "std")

# I-GARCH(1,1) with norm errors
e.spec <- ugarchspec(variance.model = list(model="iGARCH", garchOrder=c(1,1)),
                  mean.model = list(armaOrder=c(0,0)))
```

```r
# I-GARCH(1,1) with student's t errors
f.spec <- ugarchspec(variance.model = list(model="iGARCH", garchOrder=c(1,1)),
                     mean.model = list(armaOrder=c(0,0)),
                  distribution.model = "std")

# GJR-GARCH(1,1) with norm errors
g.spec <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder=c(1,1)),
                  mean.model = list(armaOrder=c(0,0)))

# GJR-GARCH(1,1) with student's t errors
h.spec <- ugarchspec(variance.model = list(model="gjrGARCH", garchOrder=c(1,1)),
                     mean.model = list(armaOrder=c(0,0)),
                  distribution.model = "std")

spec <- c(a.spec, b.spec, c.spec, d.spec, e.spec, f.spec, g.spec, h.spec)
data <- currency$logR[-1]

myFun <- function(x) {
  ugarchfit(data, spec = x)
}

# List of all models
models <- sapply(spec, myFun)
t(sapply(models, infocriteria)[1:2, ])

# Plots
chosen <- models[[4]]

estimate.dt <- currency[, .(Date, logR)]
set(estimate.dt, j = "sigma", value = c(NA, sigma(chosen)))

plot(abs(logR - coef(chosen)[["mu"]]) ~ Date, data = estimate.dt, type = "l",
     col = "lightgray", ylab = "", xlab = "Time")
lines(sigma ~ Date, data = estimate.dt, type = "l", col = "steelblue")
legend("topright", legend = c("| r_t - r |", "sigma"),
       col=c("lightgray","steelblue"), lwd=1, xpd = T)

#forecast
pred <- ugarchforecast(chosen, n.ahead=20)
```

```r
plot(tail(sigma(pred), 20),type="l",ylab="sigma",xlab="$i$",
     ylim = c(0.006, 0.008), col = "steelblue")
abline(h = sqrt(uncvariance(chosen)), lty = "dashed")

vol.20.forecast <- sqrt(sum(tail(temp, 20)^2))
```

```r
######################
# Code for Problem 2 #
######################
library(data.table)
library(ggplot2)
library(zoo)
library(lubridate)
library(sandwich)

FF <- fread("./F-F_Research_Data_Factors.CSV")
setnames(FF,c("V1","Mkt-RF"),c("date","MktRF"))
FF[,`:=`(year=date%/%100, month=date%%100)]
FF[,date:=as.yearmon(paste(year,month,sep="-"))][,`:=`(year=NULL,month=NULL)]
FF <- FF[year(date)>=1960]

industry <- fread("industry.csv")
setnames(industry,"X","date")
industry[,`:=`(year=date%/%100, month=date%%100)]
industry[,date:=as.yearmon(paste(year,month,sep="-"))][,`:=`(year=NULL,month=NULL)]
industry <- industry[year(date)>=1960]

industry <- as.data.frame(industry)

# dropping  the industries that have missing values and are reported as ???99.99
i=2
while (i<=ncol(industry)){
  if(industry[1,i]==-99.99) industry[,i]=NULL else i=i+1
}

industry <- as.data.table(industry)
setkey(FF,date)
setkey(industry,date)
reg <- merge(FF,industry)
ind_exret <- reg[,lapply(.SD, function(x) x-RF),
```

```
                 .SDcols=-c("MktRF","SMB","HML","RF"),by=date]
setkey(ind_exret,date)

reg_er <- merge(FF[,.(date,MktRF)],ind_exret)


reg_melt <- melt(reg_er,id.vars=c("date","MktRF"))

tseries_reg <- function(x){
  reg <- lm(value ~ MktRF,data=x)
  err <- sqrt(vcovHC(reg, type="HC0")[2,2])
  data.frame(t(c(coef(reg)[1],coef(reg)[2],err,summary(reg)$r.squared)))

}

betas <- reg_melt[,tseries_reg(.SD),by=variable]
setnames(betas,c("industry","alpha","beta","se","R2"))

# bar plots with error bars
ggplot(betas, aes(x=industry, y=beta)) +
  geom_bar(stat="identity", fill="steelblue", position=position_dodge()) + theme_bw() +
  geom_errorbar(aes(ymin=beta-2*se, ymax=beta+2*se),
                width=.4, position=position_dodge(.9)) + xlab("")+
  theme(axis.text.x = element_text(angle = 90, vjust = .5)) +
  theme(axis.text.x = element_text(size=16,face="plain"),
        axis.text.y = element_text(size=16,face="plain"),
        axis.title.x = element_text(size=16,face="plain"),
        axis.title.y = element_text(size=16,face="plain"),
        legend.text = element_text(size=9,face="plain"),
        legend.position = c(0.9,0.2))

# beta Statistics
min(betas$beta)
max(betas$beta)
mean(betas$beta)

# R-squared Statistics
min(betas$R2)
max(betas$R2)
mean(betas$R2)
```

```r
# scatter plot of alpha vs. beta
ggplot(betas,aes(x=beta,y=alpha)) + geom_point(size=2) +
  theme_bw() + geom_smooth(method = "lm", se = F) +
  theme(axis.text.x = element_text(size=16,face="plain"),
        axis.text.y = element_text(size=16,face="plain"),
        axis.title.x = element_text(size=16,face="plain"),
        axis.title.y = element_text(size=16,face="plain"),
        legend.text = element_text(size=9,face="plain"),legend.position = c(0.9,0.2))

# Rolling 5-year betas
beta_roll <- NA

for (i in 1:11){
  reg_sub <- reg[((i-1)*60+1):(i*60),]

  myReg <- sapply (reg_sub[,6:ncol(reg_sub)],function(x){
    mylm <- lm((x-reg_sub$RF)~reg_sub$MktRF)
    est <- summary(mylm)$coefficients[2,1]
    return(est)
  })

  beta_roll <- rbind(beta_roll,myReg)
}

beta_roll <- beta_roll[complete.cases(beta_roll),]

corr <- apply(beta_roll[,1:ncol(beta_roll)],2,function(x){
  y <- c(NA, x[-11])
  est <- cor(x[2:11],y[2:11])
  return(est)
})

corr_dt <- data.table(industry = names(corr), corr=corr)
ggplot(corr_dt, aes(x=industry, y=corr)) +
  geom_bar(stat="identity", fill="steelblue", position=position_dodge()) +
  theme(axis.text.x = element_text(angle = 90, vjust = .5)) + theme_bw() +
  xlab("") + ylab("correlations of adjacent betas")+
  theme(axis.text.x = element_text(size=16,face="plain"),
        axis.text.y = element_text(size=16,face="plain"),
        axis.title.x = element_text(size=16,face="plain"),
```

```
      axis.title.y = element_text(size=16,face="plain"),
      legend.text = element_text(size=9,face="plain"),legend.position = c(0.9,0.2))
```