Fintech HW3

Chris Hua

12/3/2016

Part 1: an OLS regression

Note that doing a regression where you lag all the predictors by 1 row is the same as a regression where you 'lead' the response variable by 1. We then regress excess return on the market vs the previous day's factors, and excluding date from the regression model.

Then, we can also calculate a few summary statistics, including the R-Squared value and p-value:

```r
lm_fit <- dd_1970 %>%
    mutate(mkt_excess = lead(mkt_excess, 1)) %>%
    select(-Date) %>%
    lm(mkt_excess ~ ., data = .)

lm_fit %>%
    summary %>%
    glance %>%
    knitr::kable(digits = 4)
```

r.squared	adj.r.squared	sigma	statistic	p.value	df
0.3564	0.2011	0.0089	2.2945	0	50

The R-squared value is about 0.36, and the p-value is rounded to 0. This is pretty good!

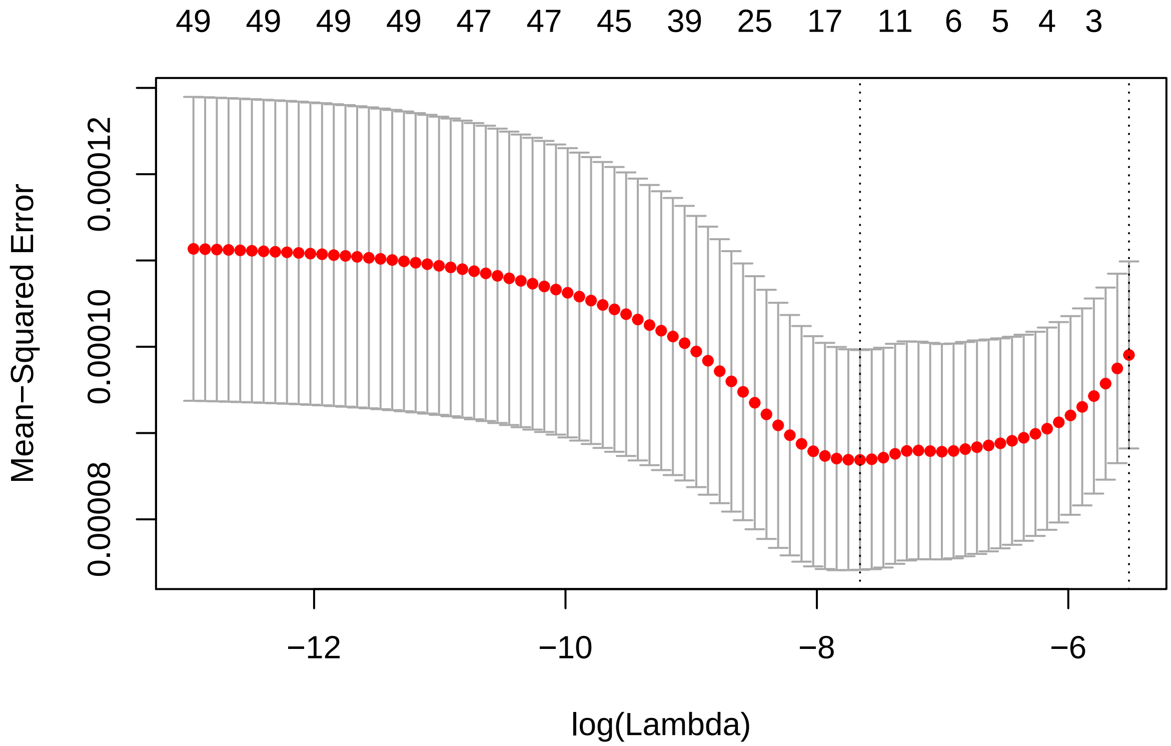
Part 2: A Lasso regression

Lasso is a regularization scheme, using the L_1 norm to 'choose' useful predictors for regression. We perform cross-validation to determine the optimal Lasso λ penalization coefficient.

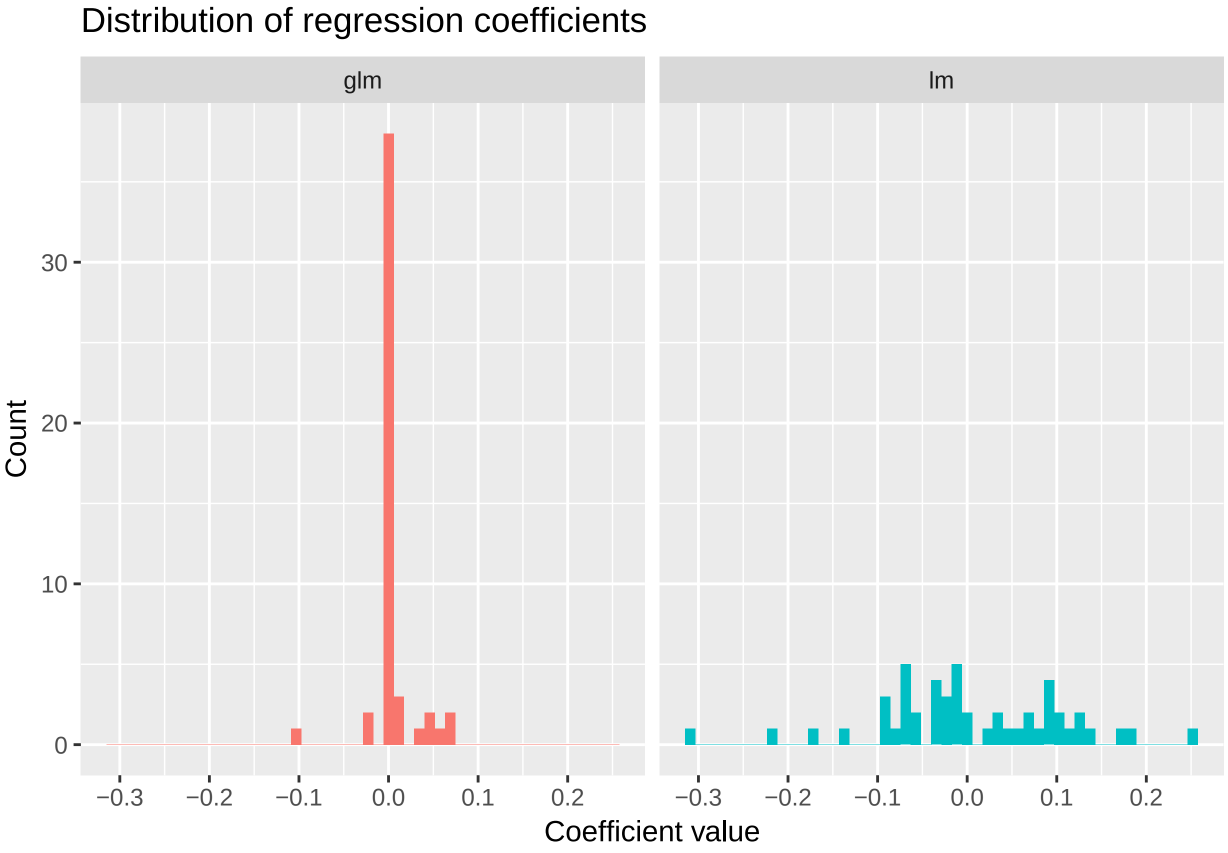
```r
lag_mat <- dd_1970 %>%
    mutate(mkt_excess = lead(mkt_excess, 1)) %>%
    select(-Date) %>%
    model.matrix(mkt_excess ~ ., data = .) %>%
    extract(,-1)

lead_y <- dd_1970 %>%
    mutate(mkt_excess = lead(mkt_excess, 1)) %>%
    select(mkt_excess) %>%
    unlist %>%
    extract(-254)

glm_fit <- cv.glmnet(x = lag_mat, y = lead_y, alpha = 1)
```



The optimal Lasso λ is at 0. Using this lambda, we can get the values of the coefficients, and then plot the distribution of the coefficients under OLS and under L1-regularization.



We notice that most of the coefficients under LASSO are centered at 0. This is because the L1-regularization draws coefficients to 0, and forces some coefficients to be 0 to create a more parsimonious model.

Part 3: Making trades, making moves

Q1: OLS

```
dd_1971 <- read.csv("Data_Daily_1971.csv")
lag_1971 <- dd_1971 %>% mutate(mkt_excess = lead(mkt_excess, 1))
```

```
pred_lm <- predict(lm_fit, lag_1971)
long_lm <- (pred_lm > 0) * 2 - 1
returns_lm <- lag_1971$mkt_excess * long_lm
returns_lm <- returns_lm[complete.cases(returns_lm)]
daily_lm <- cumprod(1+returns_lm)
```

We guess the correct direction on 61.11% of the days. Absent transaction costs, this gives us a total gross return of 0.36. The Sharpe ratio is 0.19.

Q2: Lasso

```
coef_glm2 <- as.matrix(coef(glm_fit, s="lambda.min")) %>% as.vector
mat_glm <- as.matrix(select(lag_1971, -Date))
pred_glm <- coef_glm2 %*% t(mat_glm) # yolo
long_glm <- (pred_glm > 0) * 2 -1
returns_glm <- lag_1971$mkt_excess * long_glm
returns_glm <- returns_glm[-which(is.na(returns_glm))]
daily_glm <- cumprod(1 + returns_glm)
```

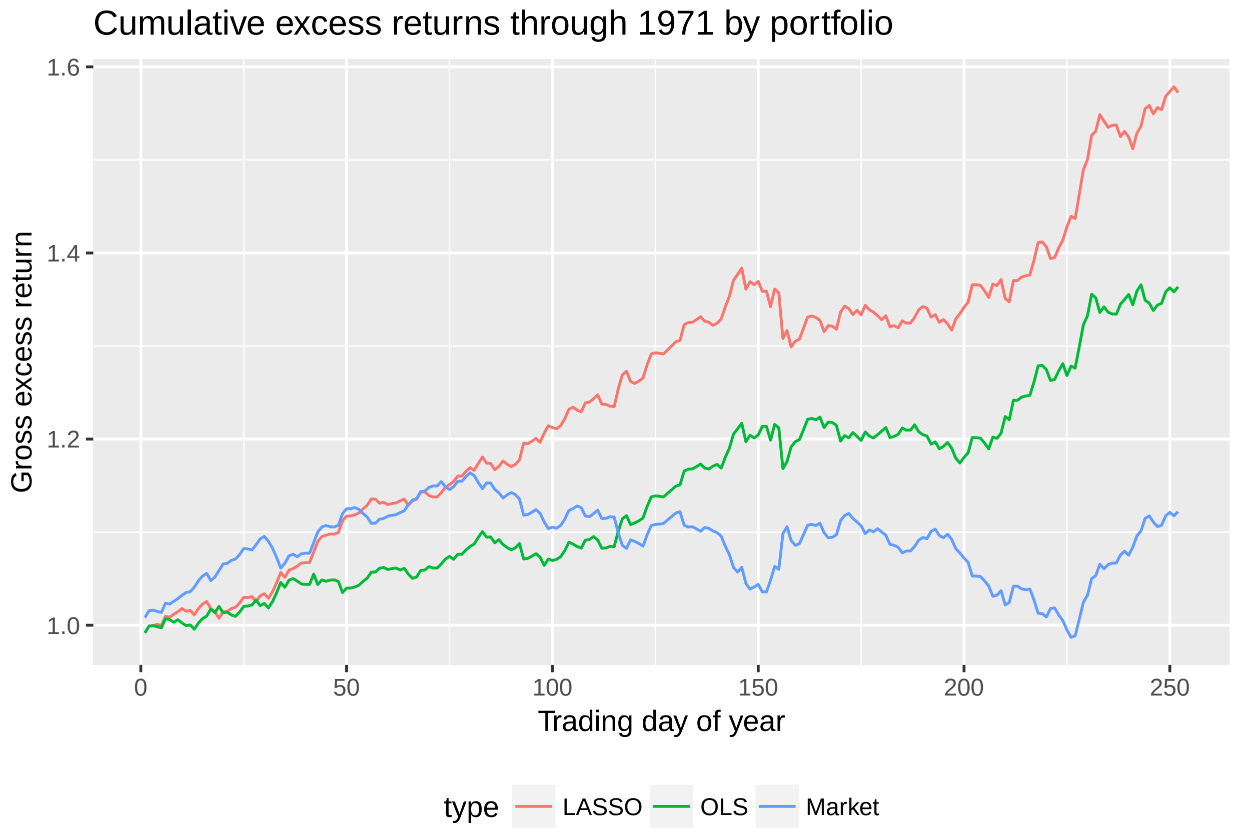
We guess correct direction on 64.29% of the days. Absent transaction costs, this gives us a total gross return of 0.57. The Sharpe ratio is 0.29.

This is significantly better than the original OLS formulation. The LASSO regularization makes the resulting model more robust to overfitting by 'drawing down' the more extreme coefficients and making the model more parsimonious by also forcing some values to 0.

We can plot these returns together:

```
df_glm_gross <- data.frame(type = "LASSO", value = daily_glm, day = 1:252)
df_lm_gross <- data.frame(type = "OLS", value = daily_lm, day = 1:252)
df_mkt_gross <- data.frame(type = "Market",
                           value = cumprod(lag_1971$mkt_excess[1:252] + 1),
                           day = 1:252)
df_gross <- rbind(df_glm_gross, df_lm_gross, df_mkt_gross)

df_gross %>%
    ggplot(aes(x = day, y= value, color = type)) +
    geom_line() + theme(legend.position = "bottom") +
    ggtitle("Cumulative excess returns through 1971 by portfolio") +
    xlab("Trading day of year") + ylab("Gross excess return")
```



The LASSO-regularized model performs the best through the year, the OLS model performs worse, and the equity portfolio performs the worst. Alpha, baby!