PSTAT 126 Final Project

Stock Portfolio Performance

Jaya Ren; Siming Su; Chloe Wang

6/12/2020

Abstract

The purpose of this project is to find out the predictors that affect most on a stock portfolio performance, based on 6 criteria that investors could be concerned about, including: annual return, excess return, systematic risk, total risk, absolute win rate, and relative win rate. By building a multiple linear regression model on the 6 criteria and 6 predictors (large B/P, large ROE, large S/P, large return rate, large market value, and small systematic risk, which will be explained in the "Data" section), we noticed there is a significant positive linear relationship between the stock relative win rate and the ROE; as well as a negative linear relationship between the win rate and large return rate, and small systematic risk with all other predictors being controlled, respectively.

Problem and Motivation

There are numerous stock choices out there in the market. How to make the best stock selection when having different preferences and criteria became a big problem for many investors. A portfolio is a grouping of financial assets such as stocks, bonds, commodities, currencies and cash equivalents, as well as their fund counterparts, including mutual, exchange-traded and closed funds. The data set we are investigating shows the performance of a stock portfolio based on different predictors from a weighted scoring stock selection model.

A weighted scoring stock selection model is one solution to the question of how to make the best stock selection. In the model, each essential criterion is given specific weights, and each stock is being scored under this system. It is a really useful tool to help determine the value of stocks in portfolios. In this project, we want to investigate which parameters are crucial to select out the best performed stocks in a given portfolio from the weighted scoring stock selection model. In that way, we are able to see a pattern for "good" stocks, and able to select them to make good profit in the investments. The definition of "good" here can vary from different criteria, which is presented as 6 different response variables(indicators) in our data set.

Data

Source: This data set is from UCI Machine Learning Repository

Our set of data has 315 instances with 12 attributes. There are 6 predictor variables to define the weights of the stock-picking concepts:

X1 - the weight of the Large Book-to-market ratio (B/P) concept Book-to-market ratio is a ratio used to find the value of a company by comparing the book value of a firm to its market value. Higher the B/P ratio, Higher is the possibility that the stock is undervalued.

X2 - the weight of the Large Return on Equity (ROE) concept Return on equity is a measure of financial performance calculated by dividing net income by shareholders' equity. It is considered a measure of how effectively management is using a company's assets to create profits.

X3 - the weight of the Large price-to-sales ratio (S/P) concept Price-to-sales ratio is calculated by taking a company's market capitalization and dividing it by the company's total sales or revenue over the past 12 months. The lower the P/S ratio, the more attractive the investment.

X4 - the weight of the Large Return Rate in the last quarter concept Rate of return is the net gain or loss of an investment over a specified time period, expressed as a percentage of the investment's initial cost. It is used to measure the profit or loss of an investment over time.

X5 - the weight of the Large Market Value concept Market value is the price an asset would fetch in the marketplace, or the value that the investment community gives to a particular equity or business.

X6 - the weight of the Small systematic Risk concept Systematic Risk is the risk inherent to the entire market. It is largely unpredictable and generally viewed as being difficult to avoid.

We also have 6 response variables to used as the indicators for the stock portfolio performance as follow: Y1 - Annual Return Y2 - Excess Return Y3 - Systematic Risk Y4 - Total Risk Y5 - Absolute Win Rate Y6 - Relative Win Rate

Questions of interest

Our goal is to be able to find the predictors that affect most on our stock portfolio performance, and their relationship to each other. In that way, we are looking forward to finding some features of well-performed stocks to improve our investment portfolio performance. So we are trying to answer the following questions by analyzing our data set: - Which predictors are crucial to predict the performance of a stock portfolio? - Is there any linear relationship between the predictors and any of the responses? - Which predictors have the most impact on stock portfolio annual return or win rates?

Regression methods

As we have more than one response variables in this data set, we want to first identify a response variable with the best possible linear regression model for this data set. Therefore, the first step would be comparing summary tables of all six response variables. After we decided which one will be the response variable we investigate in this model, we can conduct further tests, including non-constant variance test and sharpiro test to decide whether we need to transform our predictors or the response variable. Then we use the step function to determine if there is a reduced model with smaller AIC than the full model. We also want to conduct partial F tests to see if the reduced model is better than the full model. We also want to consider if interactive terms are appropriate in our model. After we confirm our final model, we want to check if there are outliers, high leverage points, or influential data points so we can further improve our model.

Now that we have all the steps planned out, we can start getting familar with our data set. By comparing 6 first order linear regression models corresponding to each response varialble (Appendix 2.), model with y1, "Annual Return", and y6, "Rel. Win Rate", have the highest $adjusted - R^2$, 0.6042 and 0.5883 respectively. However, the model with y6 has much smaller p-value for each predictor, indicating stronger linear relationships. Therefore, we decide to have y6, "Rel. Win. Rate", as our final predictor, and our full model now is y6 ~ x1 + x2 + x3 + x4 + x5 + x6.

After determining our predictor, we want to obtain a basic understanding of how each predictor is related to each other and our predictor y6. To achieve this, we made a few plots (Appendix 2.).

From the Residuals vs Fitted plot, there seem to be a funnelling pattern and a possible outlier at data point 2, so the model might violate constant variance. From the Normal Q-Q plot, the model looks right skewed, so it might violate normality. From the scatterplot, y6 seem to have strong linear relationship with x2, x4, and x6, but we are not so sure. From AV plots, all predictors seem to be useful, so further analysis is required.

Regression analysis

The code of this section is located at Appendix 3.

By running non-constant variance test, since $p = 0.065923 > 0.05$, we fail to reject the null hypothesis, so our model is constant variance. From Sharpiro test, since $p = 0.9394 > 0.05$, we fail to reject the null hypothesis, so our model is normal. Therefore, we don't need any transformation on our predictors or the response variable.

We want to use step function to see if there's any reduced model with smaller AIC. However, based on the result of step function, the best model is still our full model. We want further investigations to decide whether there's a better reduced model, since we prefer simpler model and the scatterplot indicates there might be some predictors that are better than the rest. Since our model is relatively small, we can use regsubsets() function. If we use R^2, BIC, or Mallow's C_p as our criteria, we would choose y6 ~ x2 + x4 + x6 as our new model. If we use RSS or $adjusted - R^2$ as our criteria, we would choose the full model.

To decide which one is a better model, we conduct partial F-test. Since p-value = $0.2086 > 0.05$, we fail to reject null hypothesis, so the reduced model is better. If we look at the summary table of y6 ~ x2 + x4 + x6, the $adjusted - R^2$ actually didn;t change much, but the individual p-value for each predictor has dropped a lot, indicating now we have strong linear relationships between each predictor and the response.

What about interactive terms? Do predictors relate to each other? To find the answer, we introduce a new model with interaction terms and read its summary table. Interestingly, although $adjusted - R^2$ improved by a little bit, individual t-test for each interactive term tells us that there isn't significant linear relationship between each interactive term and our response. To be more precise, we conduct partial F test again but this time with the interactive model as our full model. Since p-value=$0.1252 < 0.05$, the reduced model is better, so we keep y6 ~ x2 + x4 + x6.

Finally, we want to check if there are outliers, high leverage points, or influential points in our data. To do so, we use the function influenceIndexPlot(). From the plot, we can conclude that x=2 is an influential point because its cooks' distance is close to 1. By verifying studentized residual, x=2 has $t > 3$, so it is an outlier. By investigating leverage values, x= 2,4,6 are high leverage points. By investigating cooks' distance, x=2, 4, 6 are candidates for influential points, but since only x=2 has cooks' distance bigger than 0.5, only x=2 is an influential point. We remove x=2 first to see if it can improve our model. From summary table, our $adjusted - R^2$ increased by almost 0.16, which is great! Now we remove x=4 and x=6 to see if we can further improve our model. We can see from the summary table that $adjusted - R^2$ decreased by 0.1 and SSE didn't change much, so we decide that y6 ~ x2 + x4 + x6 is our final model with x=2 data point removed.

Conclusion

In this project, we investigate the possible relationships with multiple responses and variables in the stock market. After implanting the multiple linear regression model, we finalize our linear regression model as y6~x2 + x4 + x6. As described in the data description section, y6, x2, x4, and x6 represent Relative Win Rate, the weight of the Large ROE (rate on equity) concept, the weight of the Large Return Rate in the last quarter concept, and the weight of the Small systematic Risk concept respectively.

It turns out that the relative win rate in the financial market has a strong linear relationship with the Large ROE concept, large return rate last quarter, and systematic risk concept. Furthermore, the ROE is positively sloped, and the return rate last quarter and systematic risk are negatively sloped. This means that if a company's ROE is large, after controlling the other predictors, the chance of winning this trade will increase. Since ROE explains how effectively management is using a company's assets to create profits, a large ROE should indicate a greater win rate as what we found in the linear model.

The return rate and risk are negatively sloped, which shows financially that if the stock portfolio last quarter return rate is small or the systematic risk is relatively high, then the chance of winning this trade will decrease. As we mentioned above, the rate of return implies the net gain or loss of an investment (the investment only appears as loss for a negative rate of return value, since all rate of returns in our data set is non-negative, we assume a net gain in all circumstances). It makes sense to say that as the return rate gets lower, our win rate would decrease as well, with all other predictors being controlled. For the last predictor of systematic risk, a negative slope indicates a decrease in the chance of winning as risk get greater. This also makes sense because there is a tradeoff between risk and return in the stock market, and also it is unlikely that a stock portfolio keeps winning the trade all the time. Therefore, this project has reflected some of the properties in the stock market, and confirmed that there is a significant positive linear relationship between the stock relative win rate and the ROE; as well as a negative linear relationship between the win rate and large return rate, and small systematic risk with all other predictors being controlled, respectively. And this conclusion is obeying our common sense in the stock market as we analyzed above.

Citation

Liu, Y. C., & Yeh, I. C. Using mixture design and neural networks to build stock selection decision support systems. Neural Computing and Applications, 1-15. (Print ISSN 0941-0643, Online ISSN 1433-3058, First online: 16 November 2015, DOI 10.1007/s00521-015-2090-x)

Appendix

1. Data Preparation

```
library(readxl)
stock_portfolio_performance_data_set <- read_excel("stock portfolio performance data set.xlsx"
```

```r
attach(stock_portfolio_performance_data_set)
x1 <- stock_portfolio_performance_data_set$`Large B/P`
x2 <- stock_portfolio_performance_data_set$`Large ROE`
x3 <- stock_portfolio_performance_data_set$`Large S/P`
x4 <- stock_portfolio_performance_data_set$`Large Return Rate in the last quarter`
x5 <- stock_portfolio_performance_data_set$`Large Market Value`
x6 <- stock_portfolio_performance_data_set$`Small systematic Risk`
y1 <- stock_portfolio_performance_data_set$`Annual Return...14`
y2 <- stock_portfolio_performance_data_set$`Excess Return...15`
y3 <- stock_portfolio_performance_data_set$`Systematic Risk...16`
y4 <- stock_portfolio_performance_data_set$`Total Risk...17`
y5 <- stock_portfolio_performance_data_set$`Abs. Win Rate...18`
y6 <- stock_portfolio_performance_data_set$`Rel. Win Rate...19`
```

2. Data Exploration

```r
# investigate the annual returns versus other predictors
library(car)

full_1 <- lm(y1 ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(full_1)
```

```
##
## Call:
## lm(formula = y1 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25248 -0.05039  0.01817  0.06239  0.10587
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.770     19.315  -0.247    0.806
## x1             5.465     19.321   0.283    0.778
## x2             5.572     19.321   0.288    0.774
## x3             5.592     19.321   0.289    0.773
## x4             5.237     19.321   0.271    0.787
## x5             5.103     19.321   0.264    0.793
## x6             5.139     19.321   0.266    0.791
##
## Residual standard error: 0.0839 on 56 degrees of freedom
## Multiple R-squared:  0.6425, Adjusted R-squared:  0.6042
## F-statistic: 16.77 on 6 and 56 DF,  p-value: 5.791e-11
```

```r
full_2 <- lm(y2 ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(full_2)
```

```
##
## Call:
## lm(formula = y2 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29770 -0.06106  0.01748  0.06713  0.13945
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.686     23.140  -0.159    0.874
## x1             4.391     23.146   0.190    0.850
## x2             4.471     23.146   0.193    0.848
## x3             4.463     23.146   0.193    0.848
## x4             4.120     23.146   0.178    0.859
## x5             4.014     23.146   0.173    0.863
## x6             4.120     23.146   0.178    0.859
##
## Residual standard error: 0.1005 on 56 degrees of freedom
## Multiple R-squared:  0.5142, Adjusted R-squared:  0.4621
## F-statistic: 9.878 on 6 and 56 DF,  p-value: 2.045e-07
```

```r
full_3 <- lm(y3 ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(full_3)
```

```
##
## Call:
## lm(formula = y3 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16823 -0.06663 -0.01561  0.06147  0.28812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.290     24.114  -0.178    0.859
## x1             4.740     24.121   0.196    0.845
## x2             4.624     24.121   0.192    0.849
## x3             4.871     24.121   0.202    0.841
## x4             4.867     24.121   0.202    0.841
## x5             4.645     24.121   0.193    0.848
## x6             4.560     24.121   0.189    0.851
##
```

```
## Residual standard error: 0.1047 on 56 degrees of freedom
## Multiple R-squared:  0.2904, Adjusted R-squared:  0.2144
## F-statistic:  3.82 on 6 and 56 DF,  p-value: 0.002918
```

```
full_4 <- lm(y4 ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(full_4)
```

```
##
## Call:
## lm(formula = y4 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14002 -0.06277 -0.01432  0.04291  0.24063
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.909     21.661  -0.227    0.822
## x1             5.468     21.667   0.252    0.802
## x2             5.129     21.667   0.237    0.814
## x3             5.548     21.667   0.256    0.799
## x4             5.438     21.667   0.251    0.803
## x5             5.022     21.667   0.232    0.818
## x6             5.209     21.667   0.240    0.811
##
## Residual standard error: 0.09409 on 56 degrees of freedom
## Multiple R-squared:  0.5718, Adjusted R-squared:  0.5259
## F-statistic: 12.46 on 6 and 56 DF,  p-value: 7.259e-09
```

```
full_5 <- lm(y5 ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(full_5)
```

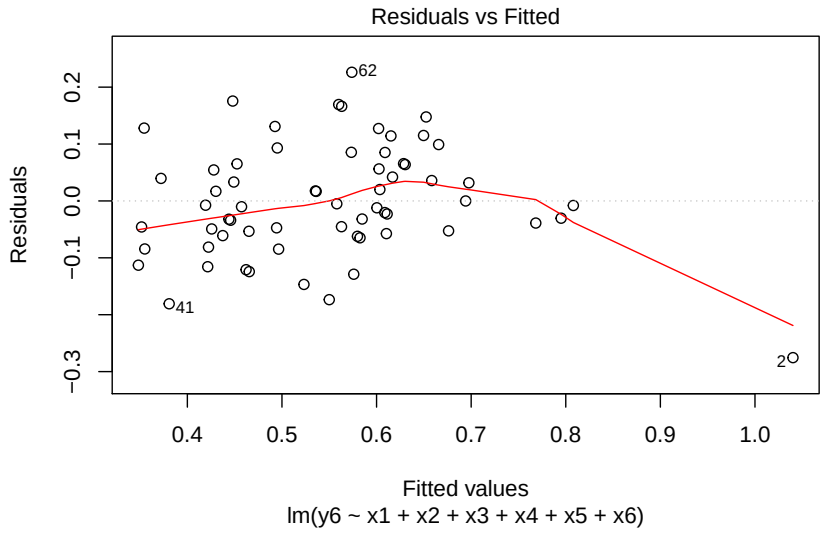
```
##
## Call:
## lm(formula = y5 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.20713 -0.05173  0.01164  0.05911  0.19903
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -14.58      21.52  -0.677    0.501
## x1             15.31      21.53   0.711    0.480
## x2             15.27      21.53   0.709    0.481
## x3             15.13      21.53   0.703    0.485
```

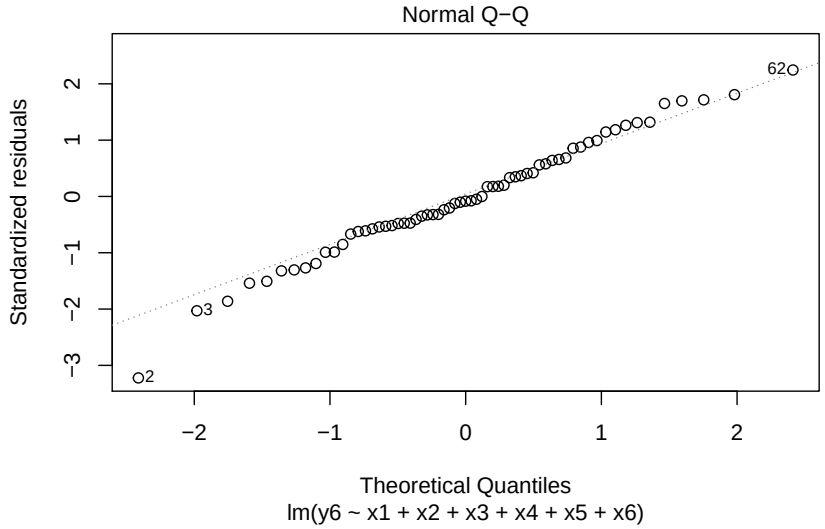
```
## x4             15.06      21.53   0.699    0.487
## x5             15.21      21.53   0.706    0.483
## x6             14.93      21.53   0.694    0.491
##
## Residual standard error: 0.0935 on 56 degrees of freedom
## Multiple R-squared:  0.3795, Adjusted R-squared:  0.313
## F-statistic: 5.709 on 6 and 56 DF,  p-value: 0.0001102
```

```
full_6 <- lm(y6 ~ x1 + x2 + x3 + x4 + x5 + x6)
summary(full_6)
```

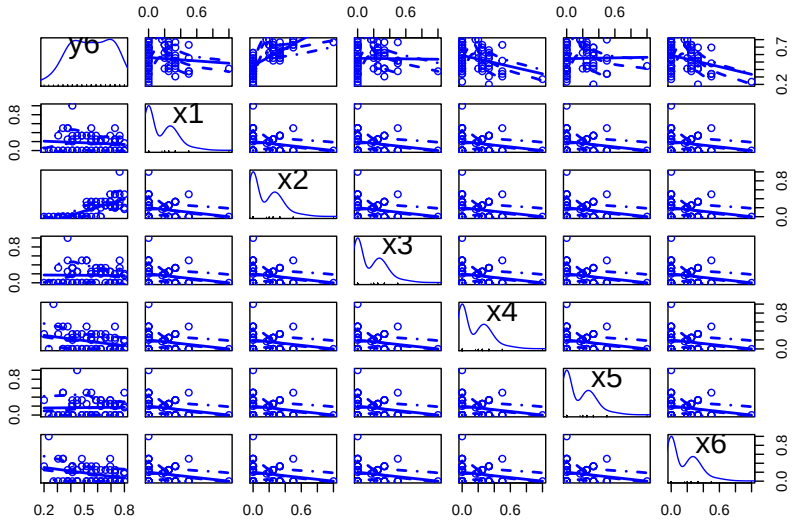
```
##
## Call:
## lm(formula = y6 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.275444 -0.055415 -0.008017  0.064498  0.226141
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -45.16      23.56  -1.917   0.0603 .
## x1             45.66      23.56   1.938   0.0577 .
## x2             46.20      23.56   1.961   0.0549 .
## x3             45.71      23.56   1.940   0.0574 .
## x4             45.52      23.56   1.932   0.0585 .
## x5             45.74      23.56   1.941   0.0573 .
## x6             45.51      23.56   1.931   0.0585 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1023 on 56 degrees of freedom
## Multiple R-squared:  0.6282, Adjusted R-squared:  0.5883
## F-statistic: 15.77 on 6 and 56 DF,  p-value: 1.668e-10
```

```
# after comparing these 6 models, full_6 is a good place to start, so we start by
# looking its diagnostic plots
plot(full_6, which = c(1,2)) # it looks a little bit right skewed in qq plot
```

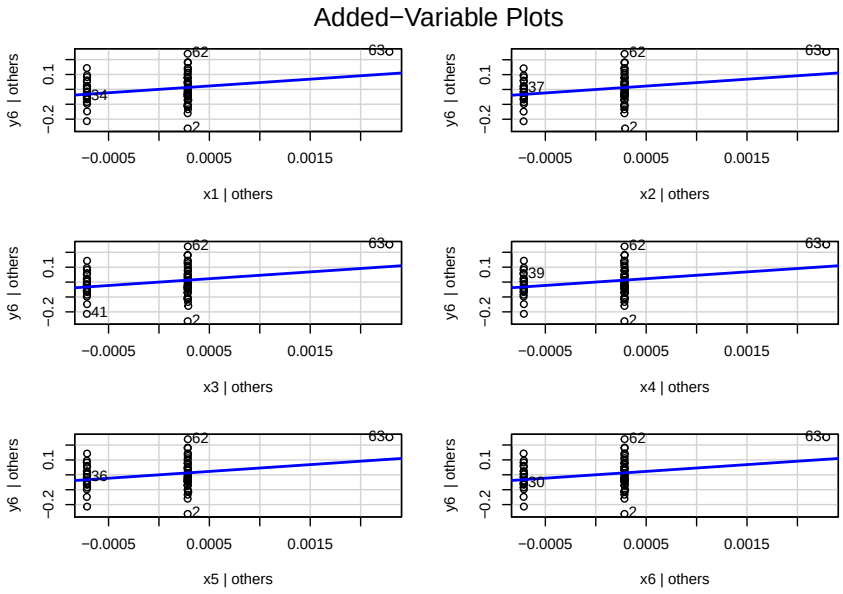





```
                        # its residual fitted plots looks like a funnelling parttern
                        # and looks like non-constant variance
scatterplotMatrix(~y6 + x1 + x2 + x3 + x4 + x5 + x6)
```




```
avPlots(full_6)
```



3. Regression Analysis Code

```
# do the test of non constant
ncvTest(full_6)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 3.381736, Df = 1, p = 0.065923
```

```r
# fail to reject, so it is constant

# do the test of noramlity
shapiro.test(full_6$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  full_6$residuals
## W = 0.99136, p-value = 0.9394
```

```r
# also fail to reject, so it is normal

# use step function to see if there is any recommonded reduced model
step(full_6, trace = 0)
```

```
##
## Call:
## lm(formula = y6 ~ x1 + x2 + x3 + x4 + x5 + x6)
##
## Coefficients:
## (Intercept)           x1           x2           x3           x4           x5
##      -45.16        45.66        46.20        45.71        45.52        45.74
##          x6
##       45.51
```

```r
# keep full model


# Investigating more with reduced model,
# let us see if anything can be improved by implenting the function regsubsets
# can say a lot in this section
library(leaps)
mod.reg <- regsubsets(cbind(x1, x2, x3, x4, x5, x6), y6)
sum.reg <- summary(mod.reg)
print(sum.reg$which)
```

```
##   (Intercept)    x1   x2    x3    x4    x5    x6
## 1        TRUE FALSE TRUE FALSE FALSE FALSE FALSE
## 2        TRUE FALSE TRUE FALSE FALSE FALSE  TRUE
## 3        TRUE FALSE TRUE FALSE  TRUE FALSE  TRUE
## 4        TRUE  TRUE TRUE  TRUE FALSE  TRUE FALSE
## 5        TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE
## 6        TRUE  TRUE TRUE  TRUE  TRUE  TRUE  TRUE
```

```r
print(sum.reg$rsq)  # choose the one increase the most, so choose #3, which are
```

```
## [1] 0.5165904 0.5485645 0.5970352 0.6033469 0.6033935 0.6281638
```

```r
                    # y6 ~ x2 + x4 + x6
print(sum.reg$rss)  # SSE, choose the smallest, choose #6, the full model
```

```
## [1] 0.7621767 0.7117641 0.6353419 0.6253904 0.6253170 0.5862624
```

```r
print(sum.reg$adjr2)  # adjusted r^2, choose the biggest, choose #6, the full model
```

```
## [1] 0.5086656 0.5335166 0.5765454 0.5759915 0.5686034 0.5883242
```

```r
print(sum.reg$cp)   # mallow's cp, choose the smallest, choose #3, which are
```

```
## [1] 13.803397 10.987970  5.688094  6.737525  8.730510  7.000000
```

```r
                    # y6 ~ x2 + x4 + x6
print(sum.reg$bic)  # BIC, choose the smallest BIC, choose #3, which are
```

```
## [1] -37.50786 -37.67593 -40.68854 -37.53999 -33.40426 -33.32407
```

```r
                    # y6 ~ x2 + x4 + x6
#####(conclusion:)
##### 3 votesf for y~ x2+x4+x6, and 2 votes for the full model, so we may choose
##### y ~ x2 + x4 +x6

# let us do anova table to verify this
red.mod <- lm(y6 ~ x2 + x4 + x6)
anova(red.mod,full_6)
```

```
## Analysis of Variance Table
##
## Model 1: y6 ~ x2 + x4 + x6
## Model 2: y6 ~ x1 + x2 + x3 + x4 + x5 + x6
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     59 0.63534
## 2     56 0.58626  3   0.04908 1.5627 0.2086
```

```r
# fail to reject, so the reduced model is a better model for us.
summary(red.mod) # check this out! A nice linear model
```

```
##
## Call:
## lm(formula = y6 ~ x2 + x4 + x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26243 -0.05753 -0.01271  0.06237  0.25205
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.52768    0.02789  18.920  < 2e-16 ***
## x2           0.49946    0.06970   7.166 1.42e-09 ***
## x4          -0.18568    0.06970  -2.664  0.00994 **
## x6          -0.19244    0.06970  -2.761  0.00767 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1038 on 59 degrees of freedom
## Multiple R-squared:  0.597,  Adjusted R-squared:  0.5765
## F-statistic: 29.14 on 3 and 59 DF,  p-value: 1.099e-11
```

```r
##interactive terms
# since our model size is small, it is probably a good time to consider some
# interactive terms
int.mod <- lm(y6 ~ x2*x4 + x2*x6 + x4*x6)
summary(int.mod)
```

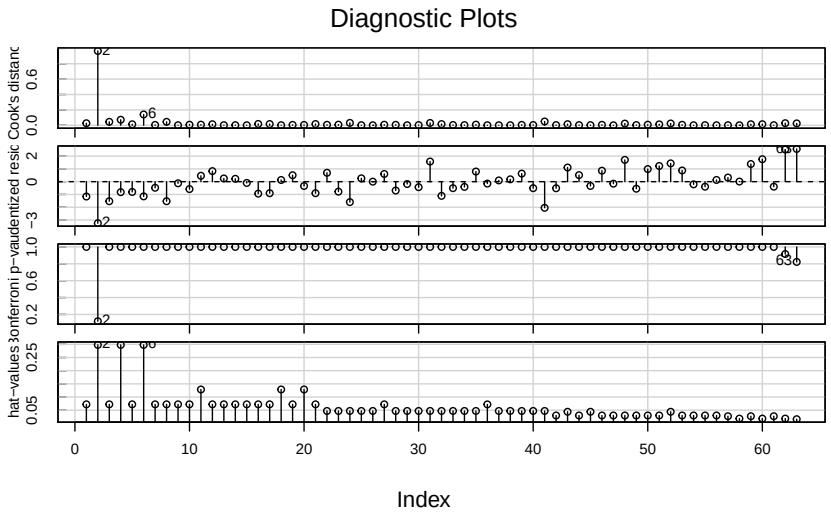
```
##
## Call:
## lm(formula = y6 ~ x2 * x4 + x2 * x6 + x4 * x6)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18896 -0.06292 -0.01205  0.06397  0.24031
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54682    0.02852  19.171  < 2e-16 ***
## x2           0.37403    0.08557   4.371 5.41e-05 ***
## x4          -0.26735    0.08557  -3.125  0.00282 **
## x6          -0.27878    0.08557  -3.258  0.00191 **
## x2:x4        0.61396    0.36144   1.699  0.09493 .
## x2:x6        0.66144    0.36144   1.830  0.07257 .
## x4:x6        0.21650    0.36144   0.599  0.55160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.1012 on 56 degrees of freedom
## Multiple R-squared:  0.6359, Adjusted R-squared:  0.5969
## F-statistic:  16.3 on 6 and 56 DF,  p-value: 9.444e-11
```

```
# interpretation: R^2 imporves only a liitle, but you add 3 more terms

# doing anova again, which one is better
anova(red.mod, int.mod)
```

```
## Analysis of Variance Table
##
## Model 1: y6 ~ x2 + x4 + x6
## Model 2: y6 ~ x2 * x4 + x2 * x6 + x4 * x6
##   Res.Df     RSS Df Sum of Sq     F Pr(>F)
## 1     59 0.63534
## 2     56 0.57399  3  0.061347 1.995 0.1252
```

```
# we compared this two models and find out that y6~x2+x4+x6 is better


## outlier detection
# reasons: the reason of doing this is because we have a relatively small R^2,
# remove some outliers may be a good choice
# first of all, see the influence plot

influenceIndexPlot(red.mod, id = TRUE)
```



```r
# in this plot, we can see that x=2 is definitely a influential point, there are
# 3 suspect high laverage points, which are x= 2, 4, 6
# 1 outlier wihch is x = 2.
# let us verify this using the rule of thumb

# outliers
rst <- abs(rstudent(red.mod))
which(rst > 2) ; which(rst > 3)
```

```
##  2 41 62 63
##  2 41 62 63
```

```
## 2
## 2
```

```r
# the result is 2 by using rule of thumb greater than 3

# high leverage
hat <- hatvalues(red.mod)
which(hat > 3*(3+1)/length(y6)); which(hat > 2*(3+1)/length(y6))
```

```
## 2 4 6
## 2 4 6
```

```
##  2  4  6 11 18 20
##  2  4  6 11 18 20
```

```r
# the result is 2,4,6 by using rule of thumb greater than 3

# influential points(cooks'distance)
cd <- cooks.distance(red.mod)
which(cd > 4/(length(y6) - 3 - 1))
```

```
## 2 4 6
## 2 4 6
```

```r
# the result is x = 2, 4, 6

# However, x = 4 and x = 6, their cook's distance is not big, so we will remove
# x=2 first and see how it goes.
stock_sub <- stock_portfolio_performance_data_set[-2,]

x2.new <- stock_sub$`Large ROE`
x4.new <- stock_sub$`Large Return Rate in the last quarter`
```

```r
x6.new <- stock_sub$`Small systematic Risk`
y6.new <- stock_sub$`Rel. Win Rate...19`
removed.red.mod <- lm(y6.new ~x2.new + x4.new + x6.new, data = stock_sub)
summary(removed.red.mod)
```

```
##
## Call:
## lm(formula = y6.new ~ x2.new + x4.new + x6.new, data = stock_sub)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18663 -0.06787 -0.01833  0.06046  0.24607
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.51254    0.02628  19.501  < 2e-16 ***
## x2.new       0.62594    0.07543   8.298 1.94e-11 ***
## x4.new      -0.18567    0.06465  -2.872  0.00569 **
## x6.new      -0.19243    0.06465  -2.977  0.00424 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09624 on 58 degrees of freedom
## Multiple R-squared:  0.6486, Adjusted R-squared:  0.6304
## F-statistic: 35.69 on 3 and 58 DF,  p-value: 3.366e-13
```

```r
## R squared improved and SSE decreases !!!!!!!!!

# let us investigate x = 4 and x = 6 to see how it went
stock_sub1 <- stock_sub[c(-4, -6),]
x2.new1 <- stock_sub1$`Large ROE`
x4.new1 <- stock_sub1$`Large Return Rate in the last quarter`
x6.new1 <- stock_sub1$`Small systematic Risk`
y6.new1 <- stock_sub1$`Rel. Win Rate...19`

removed.fur.red.mod <- lm(y6.new1 ~ x2.new1 + x4.new1 + x6.new1)
summary(removed.fur.red.mod)
```

```
##
## Call:
## lm(formula = y6.new1 ~ x2.new1 + x4.new1 + x6.new1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18348 -0.07207 -0.01067  0.06657  0.24300
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.51879    0.02743  18.913  < 2e-16 ***
## x2.new1      0.63516    0.07884   8.056 6.26e-11 ***
## x4.new1     -0.19979    0.06610  -3.022  0.00378 **
## x6.new1     -0.20655    0.06610  -3.125  0.00282 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.09657 on 56 degrees of freedom
## Multiple R-squared:  0.6483, Adjusted R-squared:  0.6295
## F-statistic: 34.41 on 3 and 56 DF,  p-value: 9.637e-13
```

```
## R squared decrease, and SSE does not change much

##### final model
# Thus, our final model will be y6 ~ x2 + x4 + x6, with x = 2 removed.
```