Name: Jiahao Chang

Project: Dogs of Dow

Course Name: AI and Machine Learning in Finance

Professor: Alok Kumar

Introduction:

The purpose of this project is to find dogs in the DOW30 portfolio. Dogs are defined as undervalued stocks that will be discovered through three traditional corporate finance approaches. R has been used to do the data analytics, and MATLAB has been used to do genetic algorithms.

The first method is the dividends per dollar that stand for the sensitivity of every unit (one dollar) change in price responding to units change in dividends payable. We will find the dividends per dollar of each stock from 2000 to 2014 (14 years in total) and assign the equal weighting to find each stock's mean dividend per dollar. The top 10 stocks that pay the highest average dividend per dollar would be named dogs; the middle ten stocks would be named others; the lowest ten would be named stars. The second method is earnings per dollar, and the third method is book value per dollar, which follows the same concept of method one but has different inputs. Each stock will be examined a fitness score after summarized scores of all other methods and rank the top 10 stocks order by highest scores (Named them as Big Dogs).

The final step is to use the genetic algorithm to generate the best dog portfolio and measure the performance of those stocks from 2015 to 2019, comparing geometric returns with an equal-weighted portfolio or a value-weighted portfolio.

Approaches and Results

	Symbol	Mean D/P
Stars	AAPL	0.676
Stars	UNH	0.781
Stars	AMGN	1.048
Stars	GS	1.144
Stars	DIS	1.168
Stars	NKE	1.174
Stars	CSCO	1.337
Stars	AXP	1.365
Stars	COKE	1.386
Stars	WBA	1.657
Others	WMT	1.766
Others	HD	1.871
Others	RTX	2.049
Others	BA	2.093
Others	TRV	2.166
Others	MSFT	2.177
Others	HON	2.239
Others	INTC	2.241
Others	IBM	2.291
Others	MMM	2.417
Dogs	MCD	2.448
Dogs	CAT	2.449
Dogs	PG	2.514
Dogs	JNJ	2.525
Dogs	JPM	2.841
Dogs	XOM	3.006
Dogs	MRK	3.491
Dogs	PFE	3.507
Dogs	CVX	3.534
Dogs	VZ	4.455

	Symbol	Mean Earn/P
Stars	COKE	0.04142801
Stars	PG	0.044237
Stars	AAPL	0.04474944
Stars	NKE	0.04548707
Stars	BA	0.04592349
Stars	MSFT	0.04723179
Stars	HON	0.04786442
Stars	AMGN	0.04840653
Stars	CSCO	0.04860679
Stars	WBA	0.04922092
Others	DIS	0.04944513
Others	MCD	0.04981885
Others	JNJ	0.05039466
Others	PFE	0.05094428
Others	HD	0.05230612
Others	MRK	0.05302659
Others	MMM	0.05334029
Others	WMT	0.05387184
Others	RTX	0.05573472
Others	VZ	0.05669828
Dogs	CAT	0.05948497
Dogs	AXP	0.06145138
Dogs	INTC	0.06290492
Dogs	XOM	0.06311484
Dogs	UNH	0.06441269
Dogs	IBM	0.06926546
Dogs	CVX	0.07258452
Dogs	JPM	0.07432359
Dogs	GS	0.08625043
Dogs	TRV	0.08667352

	Symbol	Mean Book/P
Stars	BA	9.674762
Stars	IBM	14.547588
Stars	MMM	16.420272
Stars	MSFT	17.35444
Stars	MCD	17.362719
Stars	COKE	17.9608
Stars	HD	20.440912
Stars	NKE	20.949511
Stars	JNJ	22.205416
Stars	AAPL	23.310921
Others	CAT	24.015555
Others	PG	24.261692
Others	AMGN	24.820003
Others	AXP	25.930101
Others	HON	26.185414
Others	MRK	26.206019
Others	WMT	28.137785
Others	VZ	29.432783
Others	CSCO	30.397587
Others	RTX	32.356748
Dogs	UNH	33.600779
Dogs	INTC	34.326975
Dogs	WBA	34.670027
Dogs	PFE	35.886394
Dogs	XOM	42.31601
Dogs	DIS	44.323986
Dogs	CVX	58.148759
Dogs	GS	77.357875
Dogs	TRV	80.613775
Dogs	JPM	83.936226

Figure 1

Figure 1 shows the selection of stars, others, and dogs of each method. A score of one if the stock has a Star; two if the stock has an Other; and three if the store has a Dogs. Rank the summarized scores and name them Big Stars, Big Others, and Big Dogs in ascending order (Figure 2).

Symbol	D/P SELI	Mean D/P	E/P SELECT	Mean Earn/P	B/P SELECT	Mean Book/P	Score SELECT	Score	Performance From 2015 to 2019
COKE	Stars	1.3862263	Stars	0.04142801	Stars	17.9608	Big Stars	3	199.62%
NKE	Stars	1.1741087	Stars	0.04548707	Stars	20.949511	Big Stars	3	132.64%
AAPL	Stars	0.6755262	Stars	0.04474944	Stars	23.310921	Big Stars	3	172.79%
BA	Others	2.0930307	Stars	0.04592349	Stars	9.674762	Big Stars	4	154.70%
MSFT	Others	2.1768667	Stars	0.04723179	Stars	17.35444	Big Stars	4	334.80%
AMGN	Stars	1.0475938	Stars	0.04840653	Others	24.820003	Big Stars	4	80.75%
CSCO	Stars	1.3373715	Stars	0.04860679	Others	30.397587	Big Stars	4	112.85%
MMM	Others	2.4173129	Others	0.05334029	Stars	16.420272	Big Stars	5	24.41%
HD	Others	1.8710778	Others	0.05230612	Stars	20.440912	Big Stars	5	132.66%
HON	Others	2.2394856	Stars	0.04786442	Others	26.185414	Big Stars	5	110.57%
WBA	Stars	1.6574269	Stars	0.04922092	Dogs	34.670027	Big Others	5	-10.82%
MCD	Dogs	2.4481729	Others	0.04981885	Stars	17.362719	Big Others	6	145.54%
JNJ	Dogs	2.5245385	Others	0.05039466	Stars	22.205416	Big Others	6	67.10%
PG	Dogs	2.5140192	Stars	0.044237	Others	24.261692	Big Others	6	73.49%
IBM	Others	2.2911881	Dogs	0.06926546	Stars	14.547588	Big Others	6	6.65%
WMT	Others	1.7664544	Others	0.05387184	Others	28.137785	Big Others	6	57.77%
RTX	Others	2.0490846	Others	0.05573472	Others	32.356748	Big Others	6	46.82%
AXP	Stars	1.3652957	Dogs	0.06145138	Others	25.930101	Big Others	6	66.80%
DIS	Stars	1.1676328	Others	0.04944513	Dogs	44.323986	Big Others	6	69.59%
MRK	Dogs	3.4914295	Others	0.05302659	Others	26.206019	Big Others	7	74.43%
VZ	Dogs	4.4550737	Others	0.05669828	Others	29.432783	Big Dogs	7	68.55%
UNH	Stars	0.7812981	Dogs	0.06441269	Dogs	33.600779	Big Dogs	7	198.17%
GS	Stars	1.1443827	Dogs	0.08625043	Dogs	77.357875	Big Dogs	7	43.77%
CAT	Dogs	2.4492941	Dogs	0.05948497	Others	24.015555	Big Dogs	8	116.06%
PFE	Dogs	3.5067921	Others	0.05094428	Dogs	35.886394	Big Dogs	8	50.12%
INTC	Others	2.2407929	Dogs	0.06290492	Dogs	34.326975	Big Dogs	8	108.26%
TRV	Others	2.1655071	Dogs	0.08667352	Dogs	80.613775	Big Dogs	8	48.46%
XOM	Dogs	3.0063408	Dogs	0.06311484	Dogs	42.31601	Big Dogs	9	-3.17%
CVX	Dogs	3.5337537	Dogs	0.07258452	Dogs	58.148759	Big Dogs	9	44.21%
JPM	Dogs	2.8406219	Dogs	0.07432359	Dogs	83.936226	Big Dogs	9	192.45%

Figure 2

	D/P	E/P	B/P	Scores
Stars	106.61%	136.14%	137.09%	145.58%
Others	102.51%	73.70%	80.81%	59.74%
Dogs	82.88%	82.17%	74.10%	86.69%
Dow Jone 30	62.50%	62.50%	62.50%	62.50%

Figure 3

	Trial = 3	Trial = 3	Trial = 3	Trial = 3	Trial = 3
	Crossover = 0	Crossover = 1	Crossover = 1	Crossover = 0	Crossover = 0
	Mutation Prob = 0%	Mutation Prob = 0%	Mutation Prob = 7.5%	Mutation Prob = 1.2%	Mutation Prob = 7.5%
Discontinue	0.5544	0.5747	0.5325	0.5276	0.5557
Continue	0.4887	0.4793	0.4798	0.4839	0.4836
Short + Cont	0.6224	0.6127	0.5781	0.5841	0.6393
	Use Tournament Selection				

Figure 4

I have chosen an equal-weighted performance measure to compare those four methods with Dow Jones 30 average return by holding return rate from Jan 2015 to Dec 2019. However, all the portfolios beat the Dow's average return in Figure 3, Dogs had a relatively poorer weight average performance than Stars in those five years. Figure 4 shows five settings of the genetic algorithm that I found would more likely have the highest fitness score from my project in the past: Tournament selection only, Tournament + Crossover, Crossover + Mutation Prob 7.5% + Tournament Selection, Mutation Prob 1.2%, Mutation Prob 7.5% + Tournament Selection. We will train our Dog's portfolio using continue weight + allow short (it would result in a high fitness score than discrete based on my past project.) on each dog portfolio.

Train Dogs

	Tounament = 1	Tounament = 1	Tounament = 0	Tounament = 0	Tounament = 1
	Crossover = 0	Crossover = 1	Crossover = 1	Crossover = 0	Crossover = 0
All Short+Continue	Mutation Prob = 0%	Mutation Prob = 0%	Mutation Prob = 7.5%	Mutation Prob = 1.2%	Mutation Prob = 7.5%
D/P dogs	0.2386	0.229	0.2232	0.2281	0.2234
E/P dogs	0.2811	0.2641	0.2729	0.2818	0.2752
B/P dogs	0.2658	0.2631	0.2652	0.2724	0.2575
Big dogs (score)	0.2646	0.2687	0.2676	0.2688	0.2678

Figure 5

We do experiments of each default setting three times and get the best fitness score. I was disappointed that all experiments' fitness scores were low compared to previous experiments in figure 5! Earnings per dollar method performs the best; thus, I will pick the top 2 portfolios from E/P dogs to examine performance from 2015 to 2019.

Performance from Jan 2015 to Dec 2019 (Tournament, 0.2811)											
1	AAPL	172.79%		11	HD	132.66%		21	PG	73.49%	
2	MCD	145.54%		12	CSCO	112.85%		22	TRV	48.46%	0.1154
3	MMM	24.41%		13	GS	43.77%	-0.053	23	UNH	198.17%	0.4804
4	XOM	-3.17%	-0.1183	14	MSFT	334.80%		24	COKE	199.62%	
5	RTX	46.82%		15	IBM	6.65%	-0.0605	25	VZ	68.55%	
6	PFE	50.12%		16	INTC	108.26%	0.0018	26	WBA	-10.82%	
7	MRK	74.43%		17	JNJ	67.10%		27	WMT	57.77%	
8	HON	110.57%		18	AMGN	80.75%		28	CAT	116.06%	0.3319
9	DIS	69.59%		19	JPM	192.45%	0.2038	29	BA	154.70%	
10	CVX	44.21%	0.2416	20	NKE	132.64%		30	AXP	66.80%	-0.1432
		Average return (continue weight) =				177.51%					
		Compound Annual rate =				22.65%					

Figure 6

Performance from Jan 2015 to Dec 2019 (Mutation Prob = 1.2% , 0.2818)											
1	AAPL	172.79%		11	HD	132.66%		21	PG	73.49%	
2	MCD	145.54%		12	CSCO	112.85%		22	TRV	48.46%	-0.0989
3	MMM	24.41%		13	GS	43.77%	-0.1852	23	UNH	198.17%	0.6361
4	XOM	-3.17%	-0.0943	14	MSFT	334.80%		24	COKE	199.62%	
5	RTX	46.82%		15	IBM	6.65%	-0.113	25	VZ	68.55%	
6	PFE	50.12%		16	INTC	108.26%	0.1902	26	WBA	-10.82%	
7	MRK	74.43%		17	JNJ	67.10%		27	WMT	57.77%	
8	HON	110.57%		18	AMGN	80.75%		28	CAT	116.06%	0.1991
9	DIS	69.59%		19	JPM	192.45%	0.0647	29	BA	154.70%	
10	CVX	44.21%	0.4483	20	NKE	132.64%		30	AXP	66.80%	-0.0469
		Average return (continue weight) =				185.52%					
		Compound Annual rate =				23.35%					

Figure 7

Figure 6&7 shows the performance of the top 2 E/P dogs' portfolios; I was surprised with the results. However, both results have a relatively low fitness score, both results have broken my past project's performance record (19.75% in annual compound rate). The method of earning per dollar is powerful! I noticed that every portfolio has a significant weight on UNH (UnitedHealth Group), which is a valuable stock to buy. The next step is to compare Figure 7's performance with equal-weighted and value-weighted portfolio.

Performance from Jan 2015 to Dec 2019 (Equal weight)											
1	AAPL	172.79%		11	HD	132.66%		21	PG	73.49%	
2	MCD	145.54%		12	CSCO	112.85%		22	TRV	48.46%	0.1
3	MMM	24.41%		13	GS	43.77%	0.1	23	UNH	198.17%	0.1
4	XOM	-3.17%	0.1	14	MSFT	334.80%		24	COKE	199.62%	
5	RTX	46.82%		15	IBM	6.65%	0.1	25	VZ	68.55%	
6	PFE	50.12%		16	INTC	108.26%	0.1	26	WBA	-10.82%	
7	MRK	74.43%		17	JNJ	67.10%		27	WMT	57.77%	
8	HON	110.57%		18	AMGN	80.75%		28	CAT	116.06%	0.1
9	DIS	69.59%		19	JPM	192.45%	0.1	29	BA	154.70%	
10	CVX	44.21%	0.1	20	NKE	132.64%		30	AXP	66.80%	0.1
		Average return (continue weight) =				82.17%					
		Compound Annual rate =				12.74%					

Figure 8

Performance from Jan 2015 to Dec 2019 (Value weight portfolio)														
	Symbol	Jan 2015 Price	Performnce	Weight		Symbol	Jan 2015 Price	Performnce	Weight		Symbol	Jan 2015 Price	Performnce	Weight
1	AAPL		172.79%	0.00%	11	HD		132.66%	0.00%	21	PG		73.49%	0.00%
2	MCD		145.54%	0.00%	12	CSCO		112.85%	0.00%	22	TRV	$ 101.11	48.46%	10.19%
3	MMM		24.41%	0.00%	13	GS	$ 172.41	43.77%	17.38%	23	UNH	$ 113.63	198.17%	11.46%
4	XOM	$ 85.00	-3.17%	8.57%	14	MSFT		334.80%	0.00%	24	COKE		199.62%	0.00%
5	RTX		46.82%	0.00%	15	IBM	$ 160.50	6.65%	16.18%	25	VZ		68.55%	0.00%
6	PFE		50.12%	0.00%	16	INTC	$ 33.25	108.26%	3.35%	26	WBA		-10.82%	0.00%
7	MRK		74.43%	0.00%	17	JNJ		67.10%	0.00%	27	WMT		57.77%	0.00%
8	HON		110.57%	0.00%	18	AMGN		80.75%	0.00%	28	CAT	$ 80.03	116.06%	8.07%
9	DIS		69.59%	0.00%	19	JPM	$ 61.28	192.45%	6.18%	29	BA		154.70%	0.00%
10	CVX	$ 104.98	44.21%	10.58%	20	NKE		132.64%	0.00%	30	AXP	$ 79.72	66.80%	8.04%
			Average return (continue weight) =					70.98%						
			Compound Annual rate =					11.32%						

Figure 9

Figure 8 is the performance of an equal weight portfolio, and figure 9 is the performance of the value-weighted portfolio. Both portfolios have a lower return compared with our continuous weighting portfolio. One possible insight is that the capitalization portfolio assigned too much weight on the stock that was priced high; those stocks would be challenging to rise price in percentage compared with those priced low, which can be understood as a size factor (maybe combining the concept of Fama-French could make some adjustment).

Conclusion of Dogs of Dow:

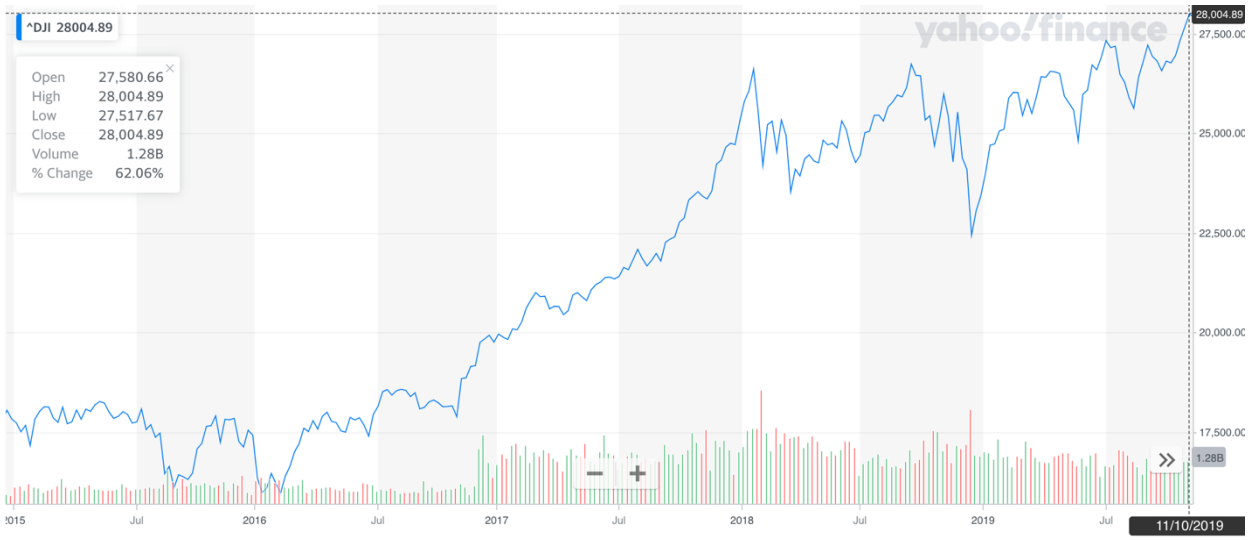


Figure 10

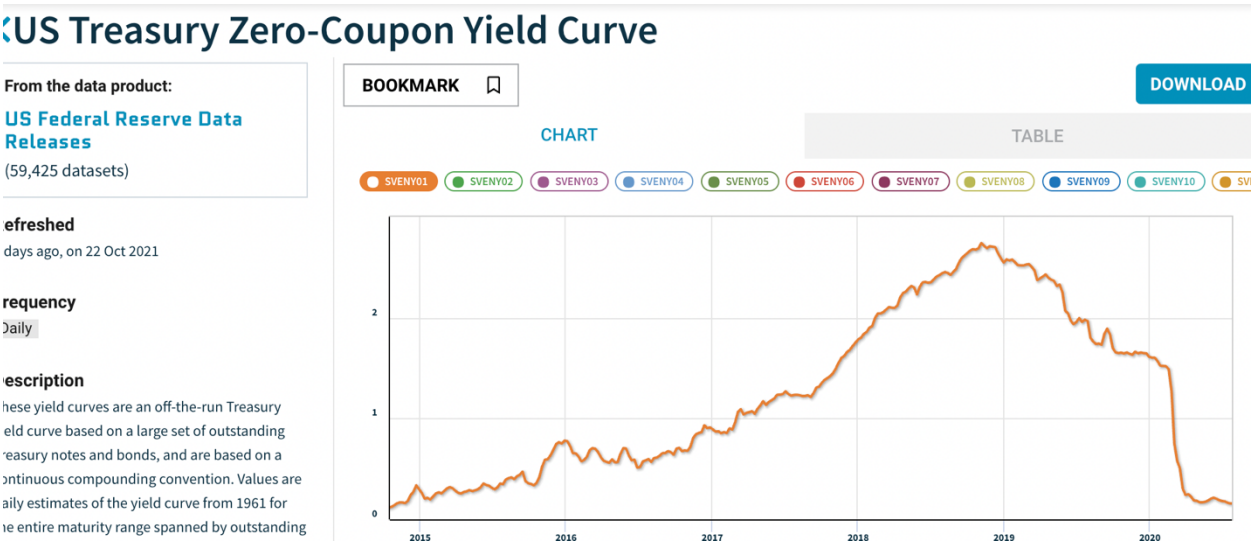


Figure 11

Despite that all shorted stocks in the continued weight with the short method result in a negative contribution to the portfolio, it still has a higher fitness score and higher performance. Shorting a stock is a hedging strategy to the portfolio; it will be effective during the recession. Figure 10 indicates Dow Jones Industry Average from January 2015 to November 2019; figure 11 marks

ten years yield curve of the US treasury zero-coupon bond (also can be seen as an inflation rate). During this period, the macro-environment is excellent, with high inflation and Dow Jone rises. Everyone has more money in their pocket. Money sometimes would fluid to the stock market in this scenario. If the macro environment is not good, that will be another story. And I believe continuing weight with the short method will still have a higher performance than discrete weight. (The idea only considered payoff but profit, you must pay a premium to short a stock due to liquidity.) "Too big to fall" is a possible answer concerning the situation that some large caps outperformed. Large caps will also carry a heavy-duty than small caps to the economy in the United States. Sometimes a slump in the industry portfolio always has a severe effect on the small company stock instead of the head of the industry. The high stock price also implied liquidity that a lot of people are watching this stock and trading it, various of parties would join the gambling in long and short positions, that may push the stock price. According to our results of four types of dogs, EP dogs did the best job, which may be the best way to find undervalued stock than the other approaches. Once a company's earnings per share are high, the company would contribute more to the investors. That company is confident with its cash flow in the future because the senior management team is creating attraction to investors in the market. That makes sense from the economic perspective.

Finding dogs of the Lithium industry in Chinese Market

Investing in stock is my interest; the Chinese stock market is different in culture than the United States stock market because that carries substantial political risk. I earned a lot of money in 2020 from investing BAIJIU (traditional Chinese alcohol, Maotai has become the stock with the highest stock price in Chinese exchange, it was 200CNY in 2013 and 2600CNY in 2021 Feb.) stock portfolio in the Chinese stock market. I realized that once most institutional investors believe this type of stock is undervalued, the stock price will increase dramatically. At the beginning of 2021, I saw an electronic car in the Chinese HUAWEI authorized phone store, and I realized that new energy vehicles would be the next BAIJIU. I went back home analyzing stocks correlated to the electronic car. HUAWEI is one of the biggest technology companies in China; HUAWEI's entering would accelerate the race of electronic cars that made China compelling with brands in other countries. During my research, I have found that several Chinese and US top private equity firms and mutual funds started to buy Lithium stocks and other raw material companies related to an electronic vehicles. I copied the portfolio of those institutions from WRDS and another Chinese database and created my lithium stock portfolio. I have earned a 50% return from March 2021 to September 2021. At the beginning of October 2021, the Federal reserve has not announced the plan (of raising the interest rate) dealing with the problem of inflation. Many institutions drop their funds from the stock market before Chinese national day (the stock exchange will take a break about seven days.) to avoid uncertainty during the seven-day holiday. My Lithium stocks price drops dramatically in my portfolio, around 30% on average.

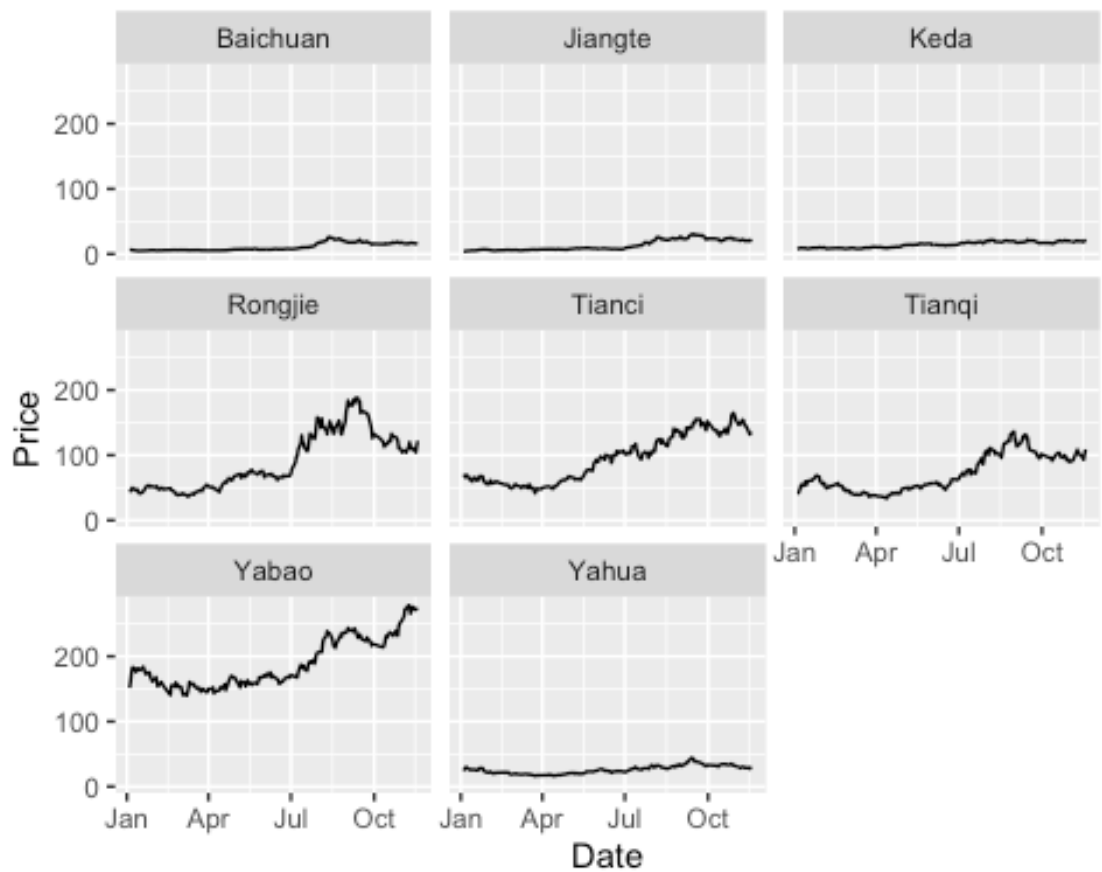
In this second part of the project, I plan to find undervalued stock from seven Chinese lithium stocks and one lithium stock from the US market; in other words, which stock is the best to buy?

The data of all those stocks are from January 2021 to November 2021, and I am focusing on short-term returns as a speculator. Those seven Chinese stocks are Tianci, Tianqi, Rongjie, Baichuan, Yahua, Jiangte, Keda. ALB is what I choose from the US exchange. Because of the time lag between US and China, Chinese investors of the lithium industry sometimes consider ALB as a reference for lithium stocks' trend on that day.

Mapping stock prices using R

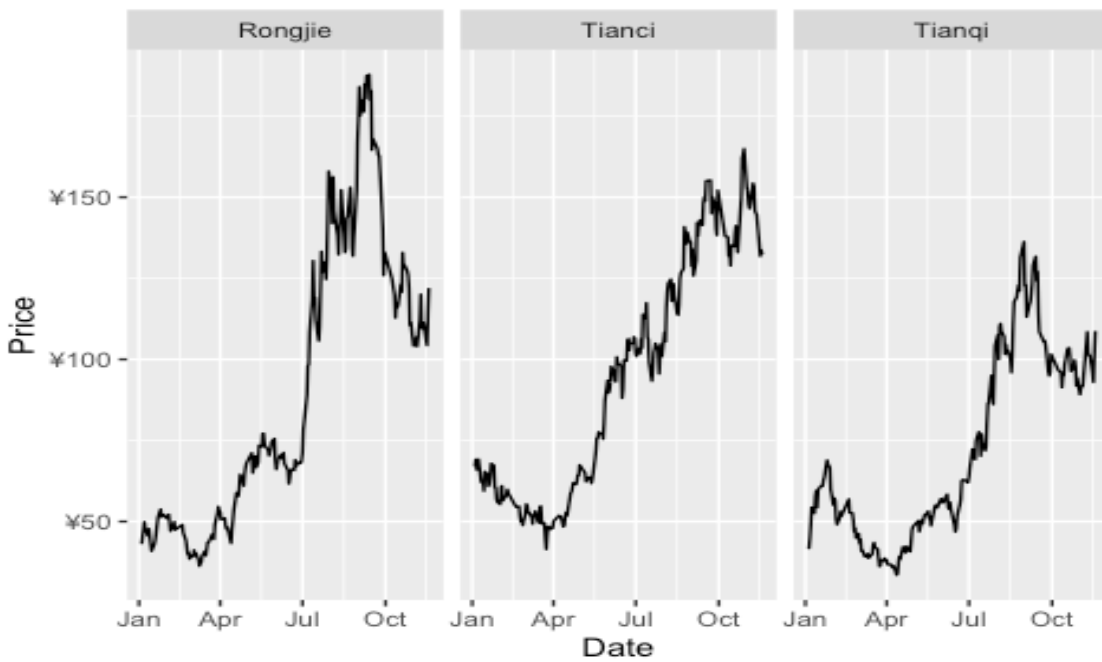
In the first step, we will graph all stocks from the portfolio—date in the X-axis and price in the Y-axis.

```
Li %>%
  ggplot( aes(x = Date, y = Price)) +
  geom_line()+
  facet_wrap(~Name)
```

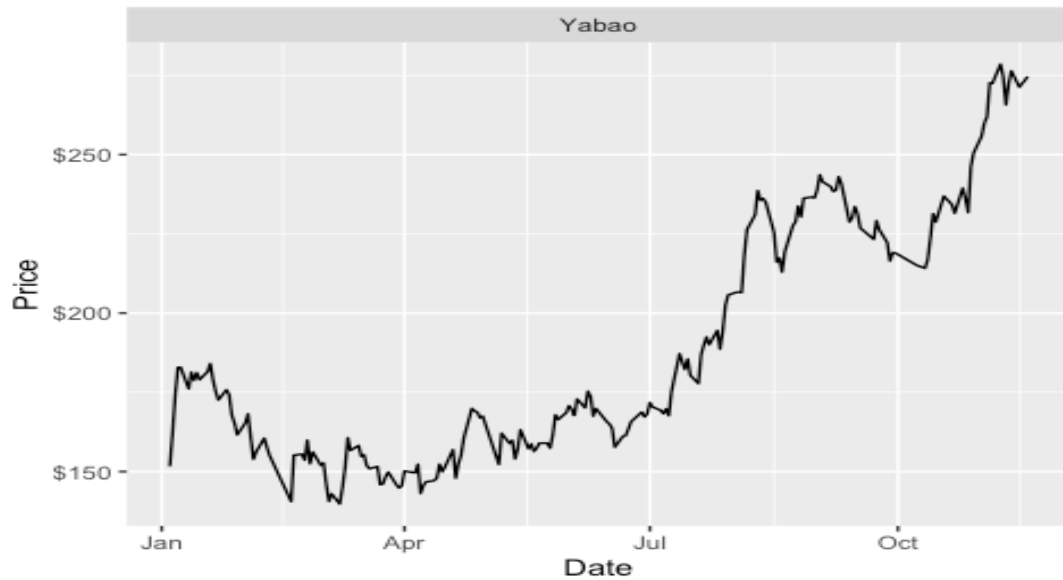


Because those stocks differ in price, it would be less appropriate to graph those stocks under the same scales. We will graph those stocks particularly.

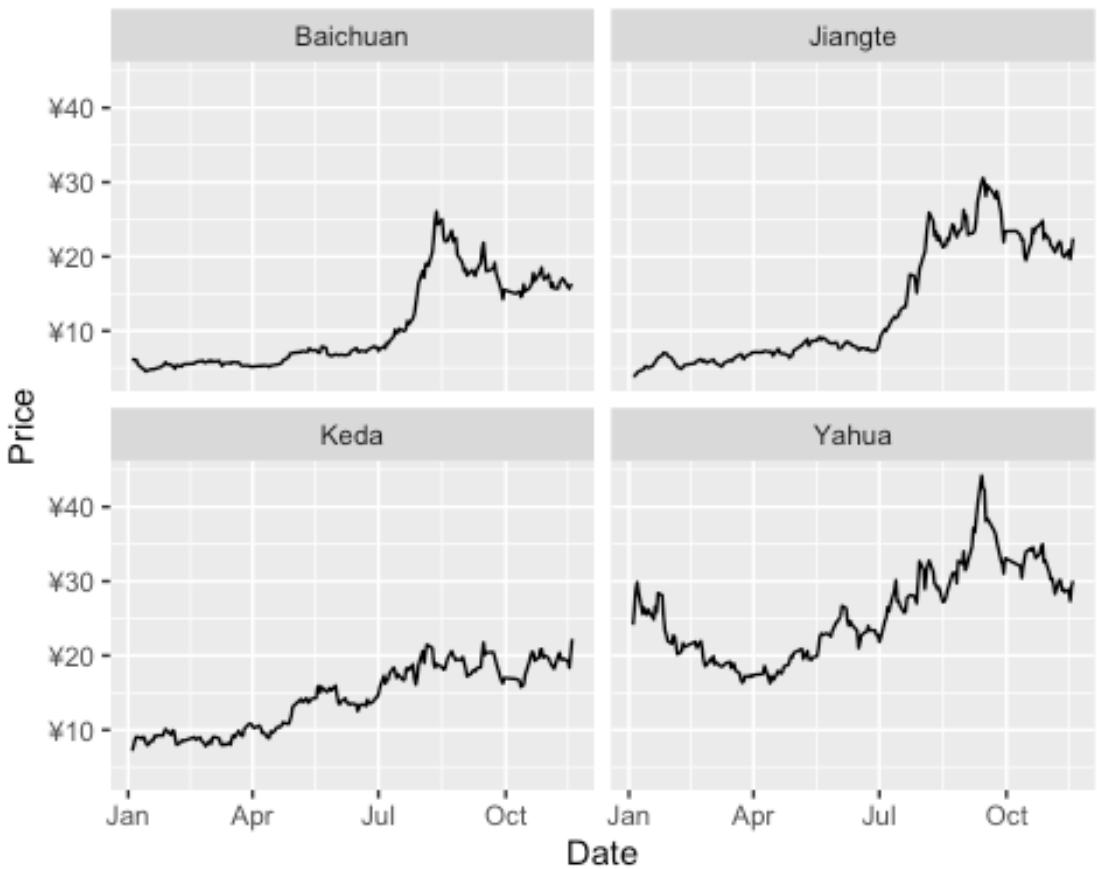
```
Big3 %>%
  ggplot( aes(x = Date, y = Price)) +
  geom_line()+
  facet_wrap(~Name)+
  scale_y_continuous(
    breaks = c(50,100,150,200),
    labels = c('¥50','¥100','¥150','¥200')
  )
```




```
Yabao %>%
  ggplot( aes(x = Date, y = Price)) +
  geom_line()+
  facet_wrap(~Name)+
  scale_y_continuous(
    breaks = c(50,100,150,200,250,300),
    labels = c('$50','$100','$150','$200','$250','$300')
  )
```




```
Sm4 %>%
  ggplot( aes(x = Date, y = Price)) +
  geom_line()+
  facet_wrap(~Name)+
  scale_y_continuous(
    breaks = c(0,10,20,30,40),
    labels = c('¥0','¥10','¥20','¥30','¥40')
  )
```



Principal Component Analysis

According to those graphs, we can see a time series pattern inside those stocks that implied a trend among those stocks. In the next step, we are going to use PCA to analyze data.

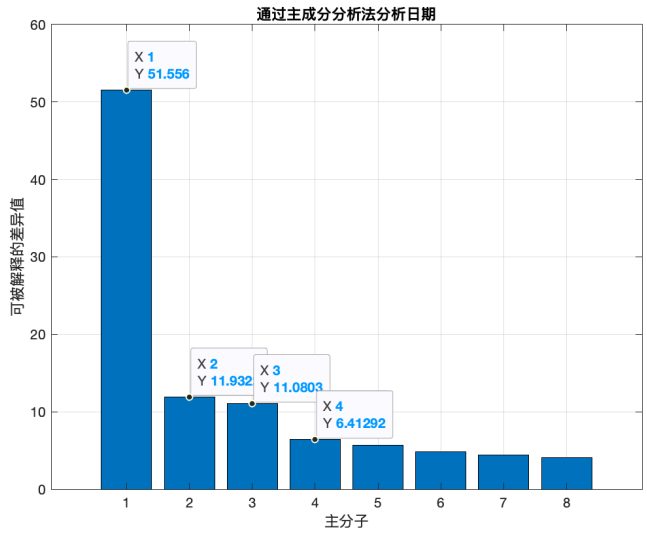


Figure 12

Figure 12 has shows PCA results, it shows the first principal component can explain 51% of variances.

Weight	0.51555965	0.11932152	0.11080253	0.06412918	0.05675328	0.04846914	0.04382081	0.04114389	
Date	PC 1	PC 2	PC 3	PC 4	PC 5	PC 6	PC 7	PC 8	Overall
Keda	0.4338573	-0.10629836	0.11370723	-0.120159	0.21466898	0.64680858	0.49277023	0.2565821	0.28101601
Baichuan	0.20926819	0.62500323	0.72808936	-0.06125187	0.04607477	-0.05605314	-0.15722463	-0.04156277	0.25222102
Rongjie	0.45453727	-0.11550753	0.04071178	0.6546808	-0.48537898	0.08701892	-0.0076609	-0.3264441	0.24338883
Tianqi	0.44075107	-0.01442052	-0.21747154	0.18020517	0.75873507	-0.25196923	-0.17282956	-0.23817075	0.23624724
Yahua	0.37830508	0.12427791	-0.20720887	0.07218637	-0.18638957	-0.33288351	-0.13577251	0.79328886	0.15887539
Tianci	0.29223674	0.46494015	-0.52827159	-0.46512379	-0.26576132	0.16124242	-0.07970526	-0.31628501	0.10702071
Jiangte	0.37026421	-0.57799769	0.29322783	-0.54596416	-0.17882058	-0.26288903	-0.15622379	-0.13881767	0.08966737
Yabao	0.03868334	0.13720793	-0.00403531	-0.03318574	-0.05635359	-0.54910358	0.80829215	-0.1432946	0.03934733

Figure 13

I have written down the weights of each stock on each principal component in Figure 13 and calculated overall importance by using a sum product of weighting of each principal component times each stock weighting inside it. And rank the overall importance at a high to low order. Based on my understanding, the overall importance is what percent of portfolio variance can be explained by one stock. For example, the highest overall importance of 28.1% of Keda that can be seen as Keda can explain 28.1% of the variance of those eight-stock portfolios; in other words, it is the best stock to buy if you believe lithium stock price will bottom out. ALB has the least overall importance, which only can explain 3.93% of the variance. From a macro perspective, the conclusion makes sense: ALB is from the US stock market; it can explain the least variance of those 7 Chinese stocks and ALB portfolios.

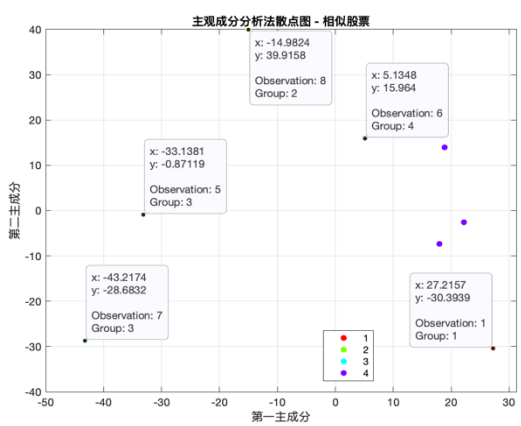


Figure 14

	X-axis	Y-axis
Explained Var	51.6%	11.9%
	Jiangte	1
	Keda	2
	Rongjie	3
	Tianqi	4
	Yabao	5
	Yahua	6
	Baichuan	7
	Tianci	8

Figure 15

Figure 14 is a clustering of those eight stocks. And figure 15 shows clustered groups of those eight stocks. 2-D overlook explained 63.5% of the variance. The finding of the clustering is that ALB (Yabao is its Chinese name, 5) is in the same cluster group of Baichuan, which may be a predictor of Baichuan stock price. Based on PCA's finding, Keda is the dog in the portfolio, and ALB may be a predictor of Baichuan.

Figure 16

Figure 17

Figures 16 and 17 are the performance of those eight stocks in two days from the App. In figure 16, Baichuan (002455) has a daily return of 3%, ALB's 3.39%, and Keda's (600499) -3.1%. In figure 17, because I took the screenshot too late, it refreshes ALB's daily return to 0; it was a 3.2% decrease on that day compared with Baichuan's -3.37%. Keda has increased by 2.64%. ALB is a good predictor of Baichuan, but we live in a multi-dimensional world; we need to do multi-factor research to decrease variance due to lurking variables; it could be another research that I will conduct in the future. I finish this part before thanksgiving break. During the break, Keda has increased 34% in four days in Figure 18 (The rule of ordinal Chinese stock has a daily limit of increase of 10%.), and I have made a lot of money of buying Keda based on my finding (I use all my money to buy Keda as a speculator.). Other lithium stocks increased dramatically, around 23% on average, but Keda increased the most. I successfully catch the DOG!

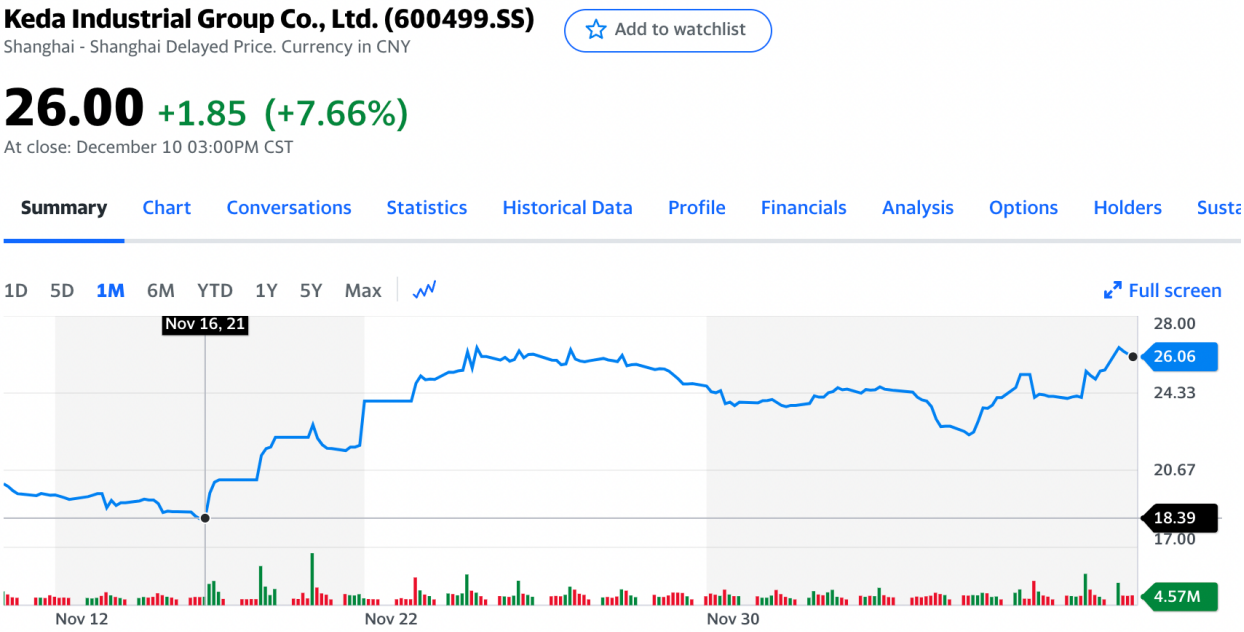


Figure 18

Appendix

Code of MATLAB (Main Function of Genetic Algorithms):

```
% Jiahao Chang, 10/12/2021, Genetic Algorithm Project.

clear all; clc;
fprintf(1, '***** Dogs of Dow with Genetic Algorithm *****.\n'); UseContWts =
0;
UseTournament = 0;
Nportf = 2000;
Ngen = 50;
MutProb = 0.012;
ContCross = 0;
AllowShort = 1;

load Dow10to14.txt;
R = Dow10to14; clear Dow10to14;
AllPermNums = unique(R(:,1)); AllDates = unique(R(:,2));

[Nmonths,Nc] = size(AllDates); [Nstocks,Nc] = size(AllPermNums); fprintf(1,
'Data read. Number of months = %d. Nstocks = %d.\n', Nmonths, Nstocks);
RR = [];
for i = 1:Nstocks

[r,c] = find(R(:,1)==AllPermNums(i,1)); RR = [RR 100*R(r,3)];
Rmkt = [100*R(r,4)];

end;

MeanR = mean(RR); StdR = std(RR); corrR = corr(RR); musigma = [MeanR; StdR];
fprintf(1, '\n\nMonthly Mean Returns and Standard Deviations:\n'); musigma
MeanMkt = mean(Rmkt); StdMkt = std(Rmkt); musigmaMkt = [MeanMkt; StdMkt];
fprintf(1, '\n\nMonthly Mean Return and Standard Deviation: MARKET\n');
musigmaMkt

if UseContWts == 0
PortfPop = GeneratePop(Nportf, Nstocks); Fitness = Fitness(PortfPop,RR);

else

PortfPop = GeneratePopWts(Nportf, Nstocks, AllowShort);

Fitness = FitnessWts(PortfPop,RR); end;

MeanFit = nanmean(Fitness);
fprintf(1, '\nAverage initial fitness (Reward-to-risk Ratio) = %.4f.\n',
MeanFit);
```

```
fprintf(1, '\nMarket fitness (Reward-to-risk Ratio) = %.4f.\n',
musigmaMkt(1)/musigmaMkt(2));

FitGen = [MeanFit]; SelPortfPop = PortfPop; for i = 1:Ngen

SelPortfPop = Crossover(SelPortfPop, Fitness, ContCross); if UseContWts == 0

MutPortfPop = Mutation(SelPortfPop,MutProb);

else

MutPortfPop = MutationCont(SelPortfPop,MutProb); end;

if UseContWts == 0
Fitness = Fitness(SelPortfPop,RR);

else

Fitness = FitnessWts(SelPortfPop,RR); end;

FitGen = [FitGen; nanmean(Fitness)];
SelPortf = Selection(Fitness,Nportf,UseTournament); SelPortfPop =
SelPortfPop(SelPortf,:);

end;
plot([0:Ngen],FitGen,'bs-','LineWidth',1.25,'MarkerSize',4); grid on;
xlabel('Generation'); ylabel('Average Fitness'); title('Mean Fitness Across
Generations');
Ntop = 1;
[SortedFitness,Idx] = sort(Fitness,'descend');

fprintf(1, '\n*** Solution + Fitness Value ***\n');
SelPortfPop(Idx(1:Ntop),:), Fitness(Idx(1:Ntop),1)'
```

Code of R (Data Visualization):

```
# Get all the data of DOW30 selection stocks

library(data.table)

d = fread("/Users/k/Desktop/dogdata.csv", header = TRUE)

d = d[,c(1,3,4,5,6,7)]

names(d) = c('Year','Symbol','BookValue','Earning','Dividend','YearEndPrice')

dow30 = subset(d, d$Symbol == 'AAPL'|d$Symbol == 'AMGN'|d$Symbol == 'AXP'|d$Symbol

== 'BA'|d$Symbol == 'CAT'|d$Symbol == 'COKE'

        |d$Symbol == 'CSCO'|d$Symbol == 'CVX'|d$Symbol == 'DIS'|d$Symbol ==

'GS'|d$Symbol == 'HD'|d$Symbol == 'HON'

        |d$Symbol == 'IBM'|d$Symbol == 'INTC'|d$Symbol == 'JNJ'|d$Symbol ==

'JPM'|d$Symbol == 'MCD'|d$Symbol == 'MMM'

        |d$Symbol == 'MRK'|d$Symbol == 'MSFT'|d$Symbol == 'NKE'|d$Symbol ==

'PG'|d$Symbol == 'TRV'|d$Symbol == 'UNH'

        |d$Symbol == 'VZ'|d$Symbol == 'WBA'|d$Symbol == 'WMT'|d$Symbol ==

'PFE'|d$Symbol == 'RTX'|d$Symbol == 'XOM')


# Three economical approach to find undervalue stocks as

dow30$Div_Over_Price = dow30$Dividend/dow30$YearEndPrice*100

dow30$Earning_Divide_Price = dow30$Earning/dow30$YearEndPrice

dow30$Book_Over_Price = dow30$BookValue/dow30$YearEndPrice*100


# Find the datas of each approach
```

```
tapply(dow30$Div_Over_Price,dow30$Symbol,mean)

tapply(dow30$Earning_Divide_Price,dow30$Symbol,mean)

tapply(dow30$Book_Over_Price,dow30$Symbol,mean)


# Selection dogs

full = fread("/Users/k/Desktop/DogofDow.csv", header = TRUE)

names(full)[3] = 'Symbol'


#Dividend per dollar dogs

DPdogs = subset(full, full$Symbol == 'MCD'|full$Symbol == 'CAT'|full$Symbol

=='PG'|full$Symbol =='JNJ'|full$Symbol =='JPM'

            |full$Symbol =='XOM'|full$Symbol =='MRK'|full$Symbol =='PFE'|full$Symbol

=='CVX'|full$Symbol =='VZ')

DPdogs = DPdogs[ , -3]

unique(DPdogs$Symbol)



# Earning per dollar dogs

EPdogs = subset(full, full$Symbol =='CAT'|full$Symbol =='AXP'|full$Symbol

=='INTC'|full$Symbol =='XOM'|full$Symbol =='UNH'

            |full$Symbol =='IBM'|full$Symbol =='CVX'|full$Symbol =='JPM'|full$Symbol

=='GS'|full$Symbol =='TRV')

EPdogs = EPdogs[ , -3]
```

```
unique(EPdogs$Symbol)



# Book value per dollar dogs
BPdogs = subset(full, full$Symbol == 'UNH'|full$Symbol =='INTC'|full$Symbol
=='WBA'|full$Symbol =='BFE'|full$Symbol =='XOM'
            |full$Symbol =='DIS'|full$Symbol =='CVX'|full$Symbol =='GS'|full$Symbol
=='TRV'|full$Symbol =='JPM')
BPdogs = BPdogs[ , -3]
unique(BPdogs$Symbol)



# Big dogs, the top 10 stocks has highest score
BigDogs = subset(full, full$Symbol == 'VZ'|full$Symbol =='UNH'|full$Symbol
=='GS'|full$Symbol =='CAT'|full$Symbol =='PFE'
             |full$Symbol =='INTC'|full$Symbol =='TRV'|full$Symbol =='XOM'|full$Symbol
=='CVX'|full$Symbol =='JPM')
BigDogs = BigDogs[ , -3]
unique(BigDogs$Symbol)



#Export data
setwd("/Users/k/Documents/R")
```

```
export(DPdogs,"DPdogs.xlsx")

export(EPdogs,"EPdogs.xlsx")

export(BPdogs,"BPdogs.xlsx")

export(BPdogs,"BigDogs.xlsx")
```