ORIE 5630 Project

Chengjian (Jason) Jin

Questions to investigate

- How would the number of assets we held short in the portfolio change when there are more and more assets in the market
- How much impact does limiting the short sell cause

Method

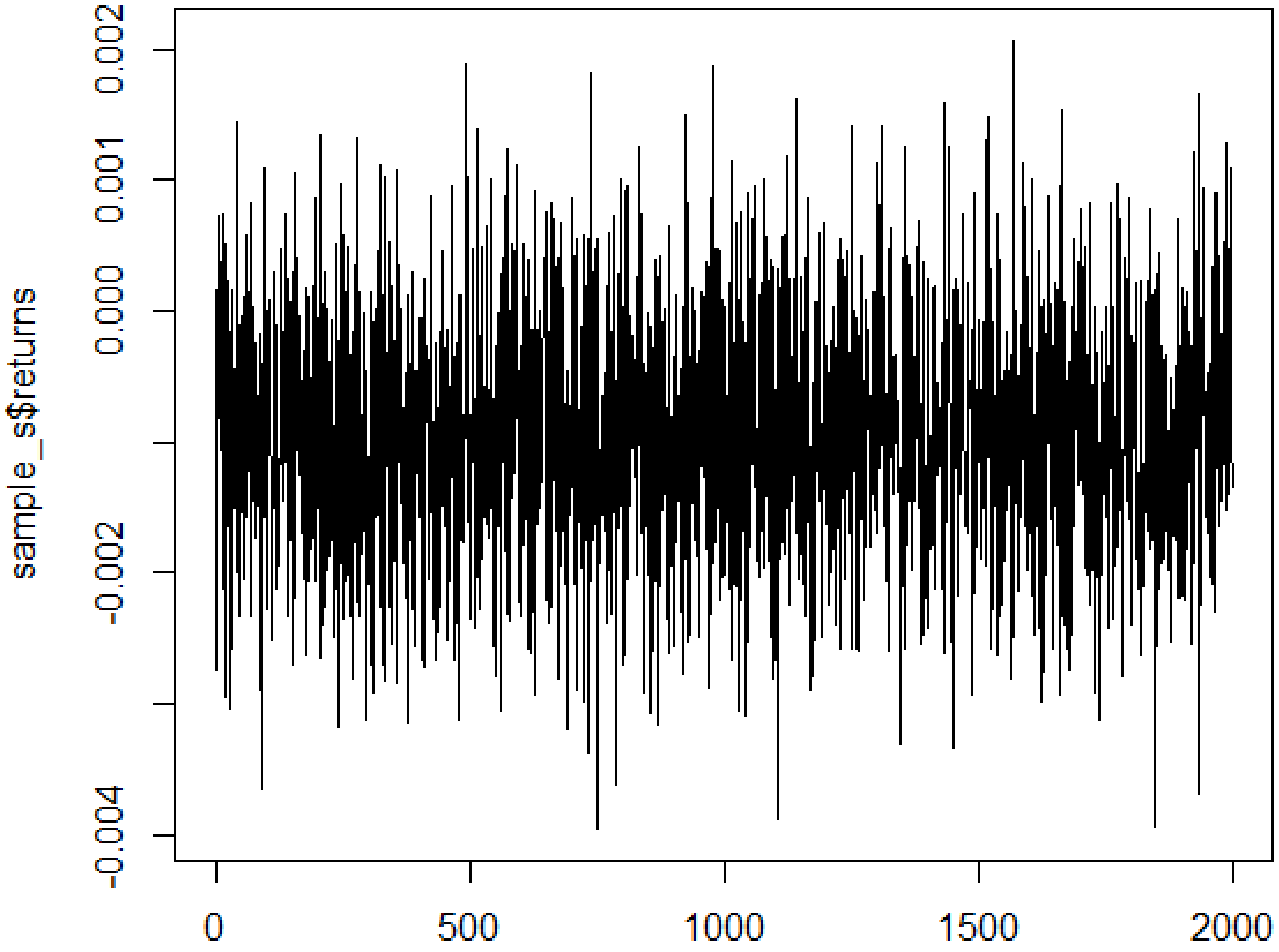
- CAPM was used for constructing the portfolio
- Simulated data was used to create stocks with known values of alpha and beta
- Used Linear Programming to maximize alpha while keeping beta value close to 0
- The result was tested again using real market data

Simulating stock returns

- Market return and risk-free return were pre-determined
- Randomly generated alpha and beta for each stock
- Stock returns were generated by the security characteristic line $R_{j,t} - \mu_{f,t} = \alpha_j + \beta_j(R_{M,t} - \mu_{f,t}) + \epsilon_{j,t},$

```r
make_stocks = function(n_stock,stock_len){
  # assign some values to the stock mean and volatility
  # mu_stock_returns = runif(n_stock,min = 0,max = 0.001)

  # simulate market return
  R_m = rnorm(stock_len,mean=0.0008,sd=0.001)
  mu_f = 0.04/253

  alpha = rnorm(n_stock,mean=0.001,sd=0.02)
  beta = rnorm(n_stock,mean=0,sd=1)

  # simulate stock returns
  R_stock = matrix(nrow=stock_len,ncol=n_stock)
  for(i in 1:stock_len){
    R_stock[i,] = alpha + mu_f + beta*(R_m[i]-mu_f)
  }

  # simulate noise for each stock
  noise = matrix(nrow=stock_len,ncol=n_stock)
  for(j in 1:n_stock){
    sigma_epsilon = runif(1,min=0,max=0.001)
    noise[,j] = rnorm(stock_len,mean=0,sd=sigma_epsilon)
  }

  # add the noise term to the stock returns
  R_stock = R_stock + noise
  R_stock = as.matrix(R_stock)

  list("returns"=R_stock,"alpha"=alpha,"beta"=beta,"R_m"=R_m)
}
```

Sample stock return plot with length 2000 days



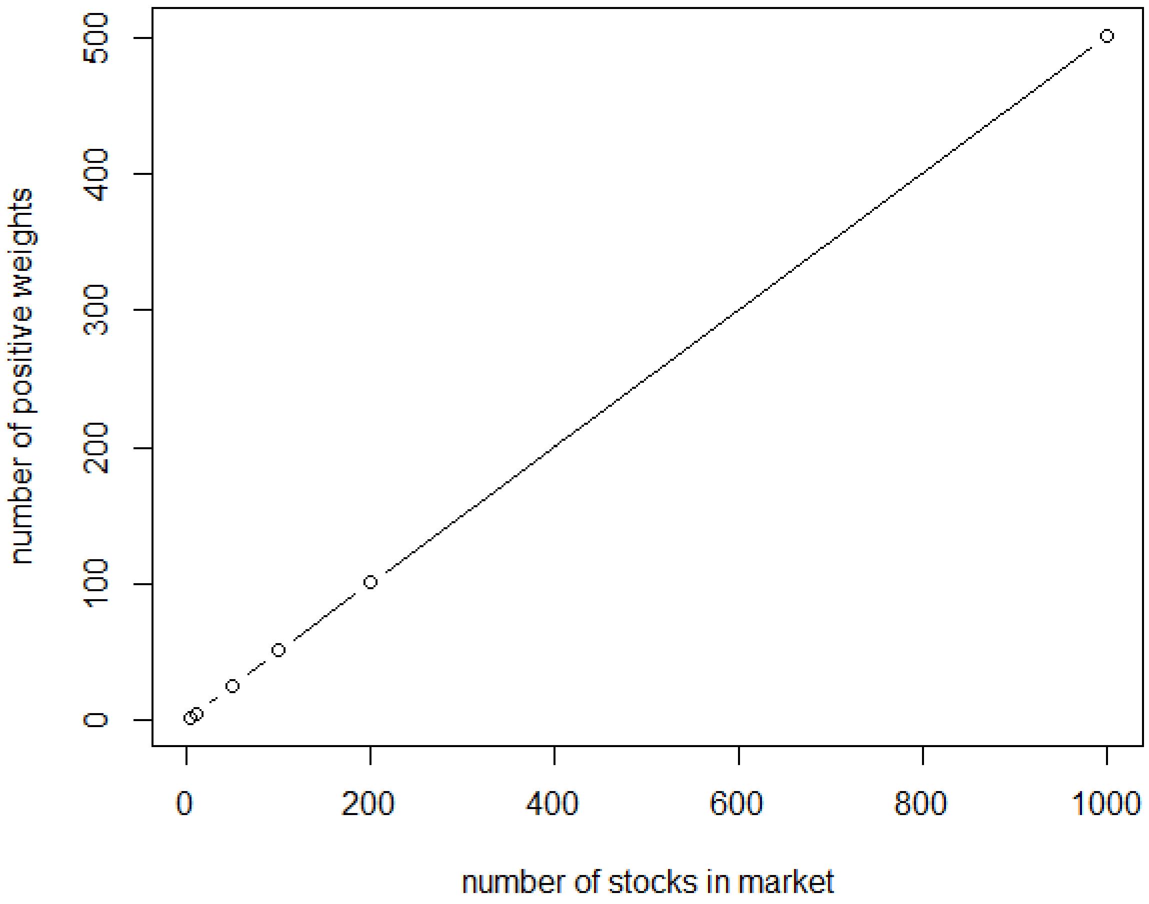
all_stocks	list [4]	List of length 4
returns	double [2000 x 1000]	1.23e-02 1.12e-02 1.26e-02 1.13e-02 1.19e-02 1.32e-02 1.52e-02 1.42e-02 ...
alpha	double [1000]	0.01225 0.01394 -0.01051 -0.05255 -0.00248 0.03273 ...
beta	double [1000]	-0.1927 0.5055 1.3284 -1.4563 -0.0499 -0.6751 ...
R_m	double [2000]	0.001610 0.000464 0.001302 0.001193 -0.000109 0.000412 ...

- An universe of 1000 stocks was created
- For a market with n stocks, select n columns from this universe

The LP to find beta neutral portfolio

```r
capm = function(stocks,n_stocks,B1,B2){
  mu_market = mean(stocks$R_m)
  alpha = stocks$alpha[1:n_stocks]
  beta = stocks$beta[1:n_stocks]
  returns = as.matrix(stocks$returns[,1:n_stocks])

  mu_stocks = colMeans(returns)

  #B1 = 1 # long limit
  #B2 = 1 # short limit

  AmatLP1 = rbind(cbind(diag(1,nrow=n_stocks),matrix(0,nrow=n_stocks,ncol=n_stocks)),cbind(matrix(0,nrow=n_stocks,ncol=n_stocks),diag(1,nrow=n_stocks)
  AmatLP2 = cbind(matrix(1,nrow=1,ncol=n_stocks),matrix(-1,nrow=1,ncol=n_stocks))
  AmatLP3 = c(beta,-beta)
  AmatLP = rbind(AmatLP1,AmatLP2,AmatLP3)

  bvec = c(rep(B1,n_stocks),rep(B2,n_stocks),1,0)
  const.dir = c(rep('<=',2*n_stocks),'=','=')
  cLP = c(alpha,-alpha)

  res_CAPM = solveLP(cvec=cLP,bvec=bvec,Amat=AmatLP,const.dir=const.dir,maximum=TRUE,lpSolve=TRUE)
  alpha_p = res_CAPM$opt

  temp1 = res_CAPM$solution[1:n_stocks]
  temp2 = res_CAPM$solution[(n_stocks+1):(2*n_stocks)]
  # temp1
  # temp2
  weights_CAPM = temp1-temp2
  return_p = sum(mu_stocks*weights_CAPM)
  # sd_p = sum(sd_returns*weights_CAPM)

  alpha_p = sum(alpha*weights_CAPM)
  beta_p = sum(beta*weights_CAPM)
  list("mean_return"=return_p,"alpha_p"=alpha_p,"beta_p"=beta_p,"w"=weights_CAPM)
}
```

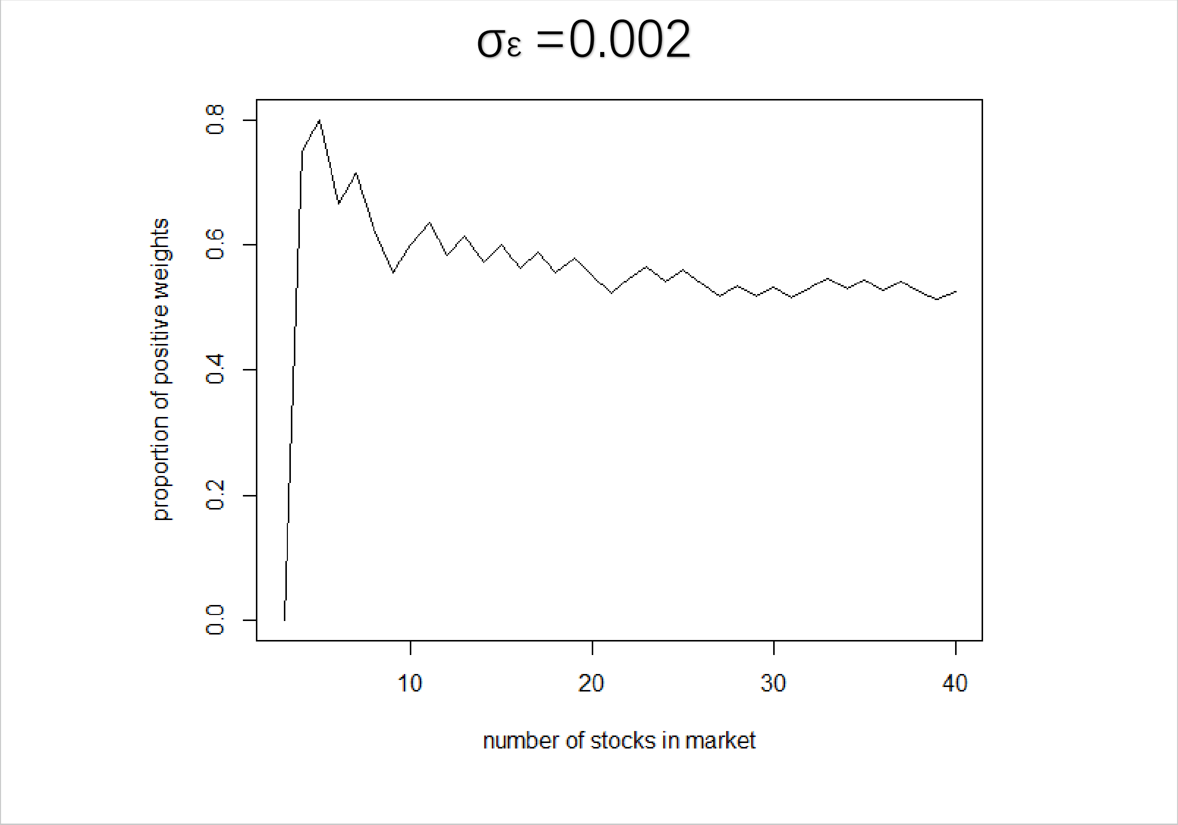
- The portfolio was created for a market with 3,10,50,100,200,1000 stocks
- The number of positive weights, portfolio alpha and beta were also printed

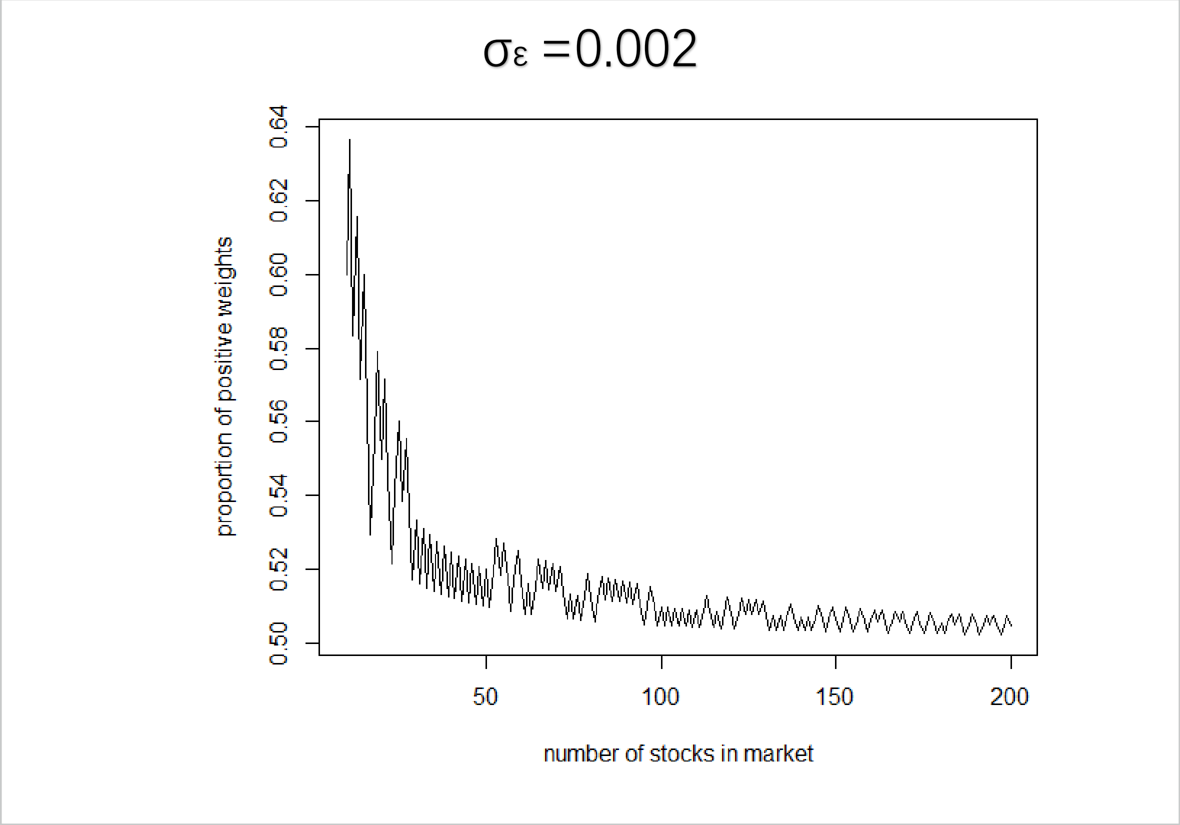



```
> positive_w
[1]    2    5   25   51  101  501
> alphas
[1]  0.02150583  0.15307513  0.63534699  1.49523681  3.08976177 16.35363628
> betas
[1]  4.197892e-13 -8.042907e-13 -3.074103e-13  3.068008e-12 -1.059946e-11  9.117835e-12
```

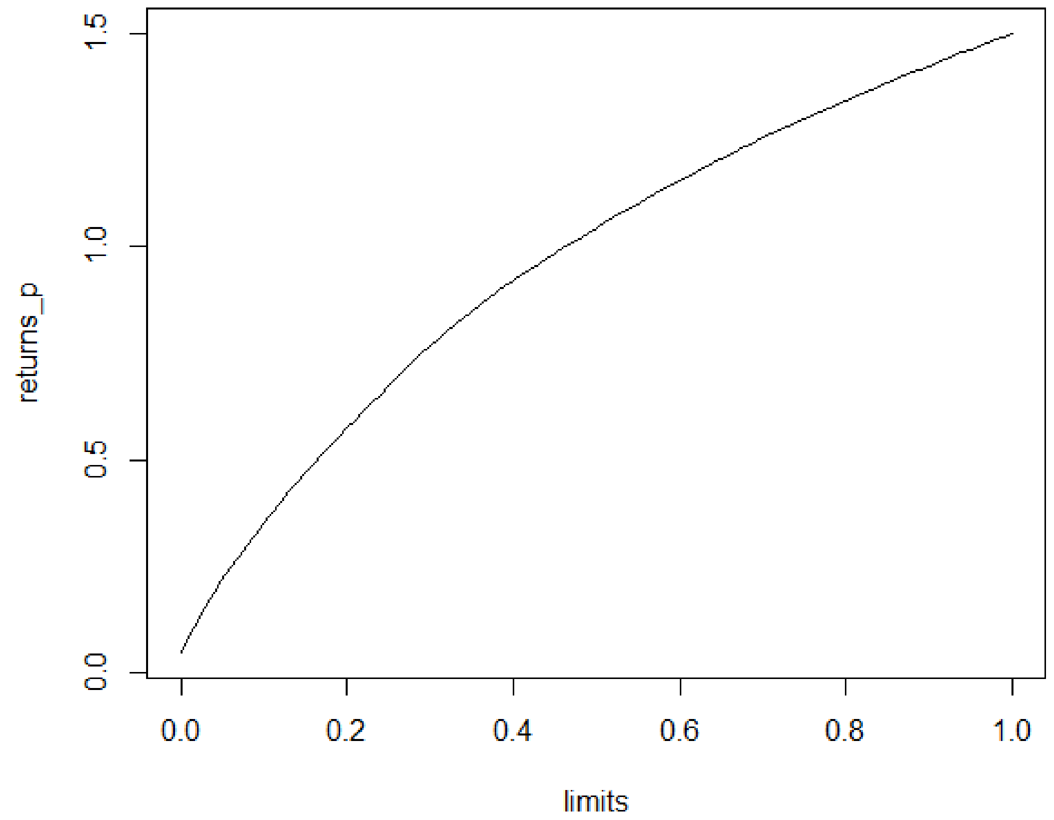
Zoom in for fewer stocks

- The proportion of positive weights converges to 50%





Effect of short sell bounds




```r
# changing the limit for short selling
limits = seq(0,1,by=0.01)
returns_p = numeric(length(limits))

weights = matrix(nrow=length(limits),ncol=100)

for(i in 1:length(limits)){
  B2 = limits[i]
  portfolio = capm(all_stocks,100,1,B2)
  weights[i,] = portfolio$w
  returns_p[i] = portfolio$mean_return
}

returns_p
plot(limits,returns_p,type='l')
```

Fix the limit

- Fixing the limit to 1 and 0
- Change the number of stocks in the market
- Huge gap between performances
- Very high cost not to have any short sell

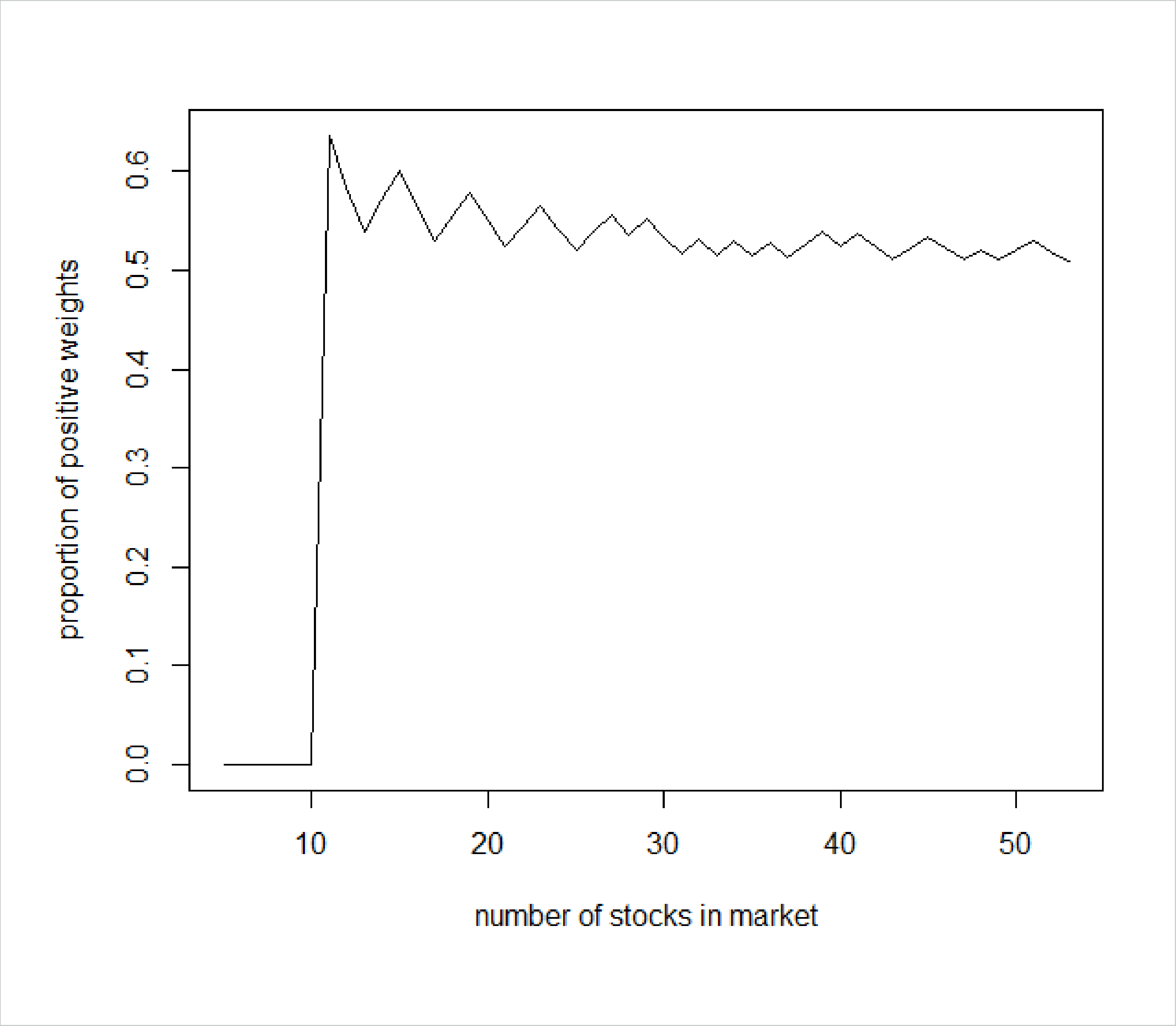
Support the argument with real data

- Total of 53 stocks from 2010 to 2021 July
- SP500 as market portfolio
- Alpha and Beta found using lm function
- Risk free rate was assumed to be 0.04/253 for simplicity

```
real_alphas = numeric(53)
real_betas = numeric(53)

sp500 = returns[,"SP500"]
mu_market = mean(sp500)
mu_stocks = colMeans(returns)

for (i in 1:53){
  fit = lm(returns[,i]~sp500)
  real_alphas[i] = fit$coefficients[1]
  real_betas[i] = fit$coefficients[2]
}
```

Weights in portfolio

- Same result as simulated data, converge to 50% in the long run



Limiting short sell

Conclusion

- As the market contains more and more stocks, the CAPM portfolio will tend to short half of the stocks
- Short sell is very important to portfolio expected return

- Reference: Levy, M and Ritov, Y, 2001, *portfolio optimization with many assets*