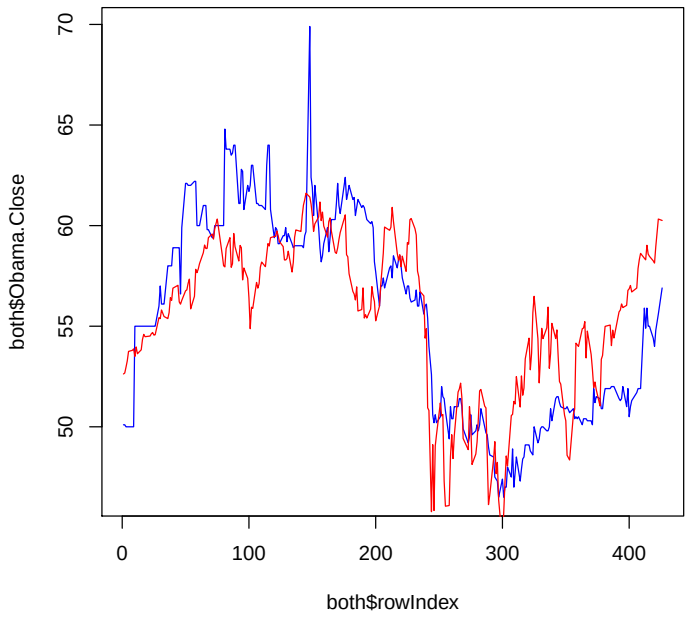
# Obama vs the SP500

First the data itself:

```
> obama <- read.csv("obama.csv")
> sp500 <- read.csv("sp500.csv")
> obama$rowIndex <- 1:(dim(obama)[1])
> both <- merge(sp500, obama, "Date")
> both <- both[sort.list(both$rowIndex), ]
```

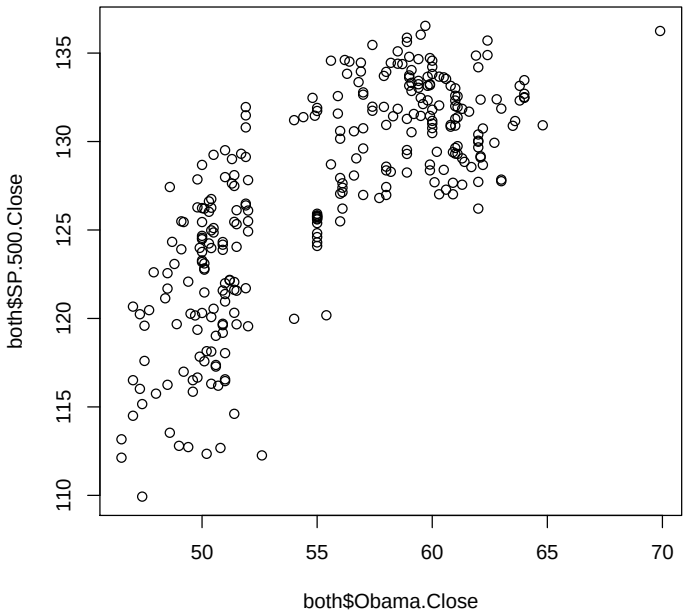
Basic plots:

```
> plot(both$Obama.Close ~ both$rowIndex, col = "blue", type = "l")
> Q <- lm(both$Obama.Close ~ both$SP.500.Close)
> alpha <- Q$coefficients[1]
> beta <- Q$coefficients[2]
> lines(alpha + beta * both$SP.500.Close ~ both$rowIndex, col = "red")
```




Or as a "X" vs "Y" plot:

```
> plot(both$Obama.Close, both$SP.500.Close)
```

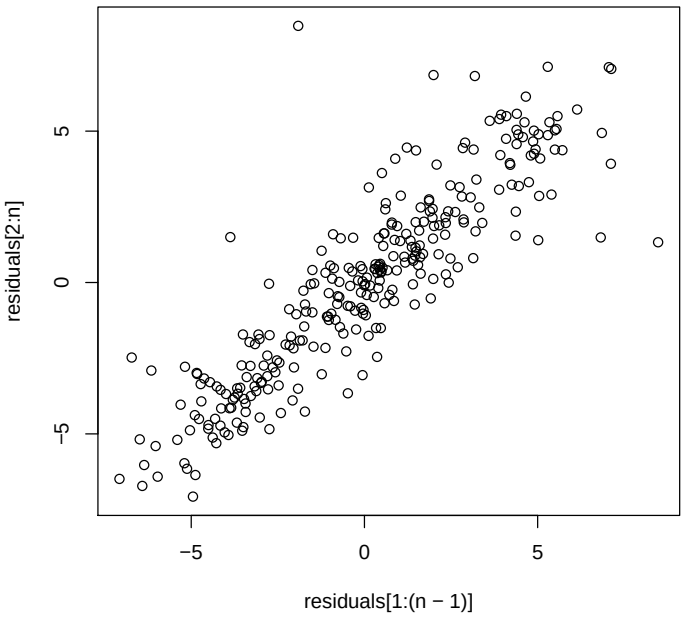



What does naive statistics think of this?

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | -27.4229 | 4.0477 | -6.78 | 0.0000 |
| both$SP.500.Close | 0.6521 | 0.0318 | 20.49 | 0.0000 |

How about issues of auto correlation?

```
> residuals = simple.model$residuals
> n = length(residuals)
> plot(residuals[2:n] ~ residuals[1:(n - 1)])
```




Yikes!

How about looking at differences (aka returns) The "lag" operator in R sucks. So let's write our own.

```
> mylag <- function(x) {
+     return(c(NA, x[1:(length(x) - 1)]))
+ }
> mylag(c(1, 2, 3, 4))

[1] NA  1  2  3

> mylag <- function(x, number.of.lags = 1) {
+     return(c(rep(NA, number.of.lags), x[1:(length(x) - number.of.lags)]))
+ }
> mylag(c(1, 2, 3, 4, 5, 6, 7, 8, 9), 3)

[1] NA NA NA  1  2  3  4  5  6

> mylag(c(1, 2, 3, 4))

[1] NA  1  2  3

> make.returns <- function(x) {
+     (x - mylag(x))/mylag(x)
+ }
> make.returns(c(1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6))
```

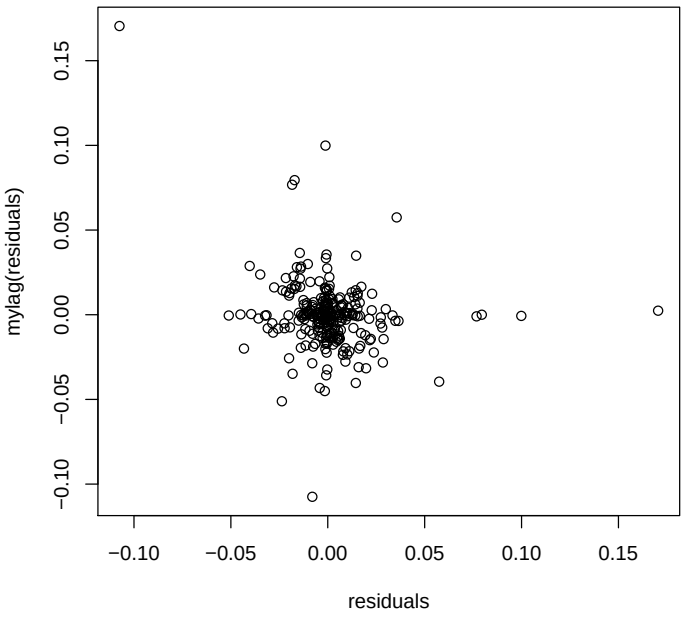
```
[1]          NA 0.10000000 0.09090909 0.08333333 0.07692308 0.07142857 0.06666667
```

```
> both$Obama.returns = make.returns(both$Obama.Close)
> both$SP.returns = make.returns(both$SP.500.Close)
> both$Obama.diffs = both$Obama.Close - mylag(both$Obama.Close)
> both$SP.diffs = both$SP.500.Close - mylag(both$SP.500.Close)
> returns.model <- lm(Obama.returns ~ SP.returns, data = both)
```

Which generates a much less impressiv regression: But good residuals:

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 0.0006 | 0.0012 | 0.50 | 0.6164 |
| SP.returns | 0.1017 | 0.0876 | 1.16 | 0.2464 |

```
> residuals = returns.model$residuals
> n = length(residuals)
> plot(mylag(residuals) ~ residuals)
```




## Is Obama predictable?

Will previous returns on the SP predict Obama returns?

```
> xtable(summary(lm(Obama.returns ~ SP.returns + mylag(SP.returns),
+     data = both)))
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 0.0005 | 0.0012 | 0.43 | 0.6645 |
| SP.returns | 0.1200 | 0.0876 | 1.37 | 0.1720 |
| mylag(SP.returns) | 0.1854 | 0.0876 | 2.12 | 0.0352 |

How about the previous week of returns:

```
> xtable(summary(lm(Obama.returns ~ SP.returns + mylag(SP.returns) +
+     mylag(SP.returns, 2) + mylag(SP.returns, 3) + mylag(SP.returns,
+     4), data = both)))
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 0.0005 | 0.0012 | 0.39 | 0.6992 |
| SP.returns | 0.1310 | 0.0894 | 1.47 | 0.1438 |
| mylag(SP.returns) | 0.1912 | 0.0894 | 2.14 | 0.0333 |
| mylag(SP.returns, 2) | -0.0351 | 0.0891 | -0.39 | 0.6939 |
| mylag(SP.returns, 3) | 0.1164 | 0.0897 | 1.30 | 0.1958 |
| mylag(SP.returns, 4) | 0.1306 | 0.0896 | 1.46 | 0.1463 |

How about the previous two weeks of returns:

```
> xtable(summary(lm(Obama.returns ~ SP.returns + mylag(SP.returns) +
+     mylag(SP.returns, 2) + mylag(SP.returns, 3) + mylag(SP.returns,
+     4) + mylag(SP.returns, 5) + mylag(SP.returns, 6) + mylag(SP.returns,
+     7) + mylag(SP.returns, 8) + mylag(SP.returns, 9) + mylag(SP.returns,
+     10), data = both)))
```

## Looking at the previous returns

```
> both$SP.average.return = (mylag(both$SP.returns) + mylag(both$SP.returns,
+     2) + mylag(both$SP.returns, 3) + mylag(both$SP.returns, 4) +
+     mylag(both$SP.returns, 5) + mylag(both$SP.returns, 6) + mylag(both$SP.returns,
+     7) + mylag(both$SP.returns, 8) + mylag(both$SP.returns, 9) +
+     mylag(both$SP.returns, 10))/10

> xtable(summary(lm(Obama.returns ~ SP.average.return, data = both)))
```

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 0.0000 | 0.0012 | 0.03 | 0.9745 |
| SP.returns | 0.1267 | 0.0875 | 1.45 | 0.1487 |
| mylag(SP.returns) | 0.1800 | 0.0873 | 2.06 | 0.0403 |
| mylag(SP.returns, 2) | -0.0033 | 0.0876 | -0.04 | 0.9699 |
| mylag(SP.returns, 3) | 0.1107 | 0.0883 | 1.25 | 0.2111 |
| mylag(SP.returns, 4) | 0.1065 | 0.0883 | 1.21 | 0.2289 |
| mylag(SP.returns, 5) | 0.0759 | 0.0894 | 0.85 | 0.3965 |
| mylag(SP.returns, 6) | -0.0731 | 0.0883 | -0.83 | 0.4084 |
| mylag(SP.returns, 7) | 0.1152 | 0.0883 | 1.31 | 0.1930 |
| mylag(SP.returns, 8) | 0.1028 | 0.0878 | 1.17 | 0.2423 |
| mylag(SP.returns, 9) | -0.1314 | 0.0876 | -1.50 | 0.1351 |
| mylag(SP.returns, 10) | 0.1977 | 0.0878 | 2.25 | 0.0252 |

| | Estimate | Std. Error | t value | Pr(>\|t\|) |
|---|---|---|---|---|
| (Intercept) | 0.0001 | 0.0012 | 0.08 | 0.9341 |
| SP.average.return | 0.6651 | 0.3246 | 2.05 | 0.0414 |

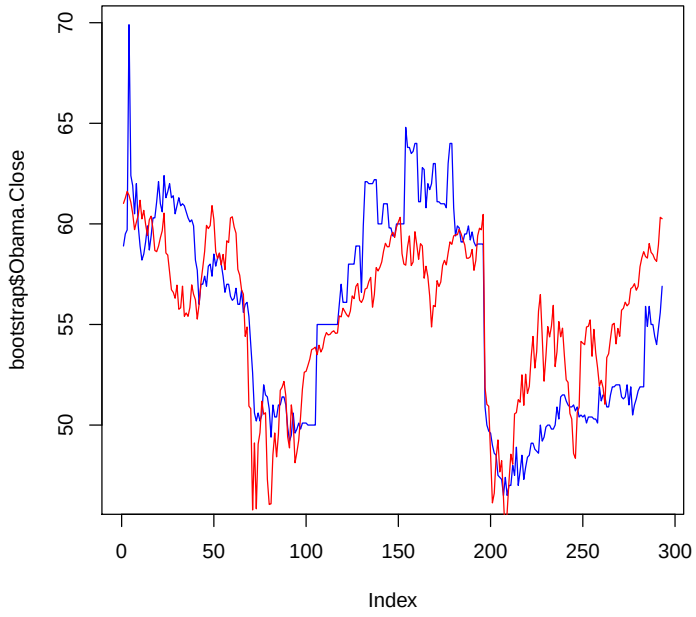
## Time series bootstrap

Let's chop this up into 20 segments (about one month each) and bootstrap these months. To do this, we will first write a block bootstrap sampler. It will take a set of blocks and make up the index file for a new data set.

```
> block_sample <- function(bins) {
+     values <- unique(bins)
+     num.values <- length(values)
+     which.bins <- sample(num.values, num.values, replace = TRUE)
+     result <- c()
+     for (i in 1:num.values) {
+         result <- c(result, (1:length(bins))[bins == values[which.bins[i]]])
+     }
+     return(result)
+ }
> block_sample(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
+     2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3))

 [1]  1  2  3  4  5  6  7  8  9 10 21 22 23 24 25 26 27 28 29 30 11 12 13 14 15
[26] 16 17 18 19 20

> block_sample(c(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
+     2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3))

 [1]  1  2  3  4  5  6  7  8  9 10 21 22 23 24 25 26 27 28 29 30 21 22 23 24 25
[26] 26 27 28 29 30
```

Let's look at what a sample pair of SP and Obama might look like using this setup:

```
> number.bins <- 3
> bins <- trunc(number.bins * (1:dim(both)[1] - 1)/dim(both)[1])
> indexes <- block_sample(bins)
> bootstrap <- both[indexes, ]
> boot.model <- lm(bootstrap$Obama.Close ~ bootstrap$SP.500.Close)
> plot(bootstrap$Obama.Close, col = "blue", type = "l")
> alpha <- boot.model$coefficients[1]
> beta <- boot.model$coefficients[2]
> lines(alpha + beta * bootstrap$SP.500.Close, col = "red")
```



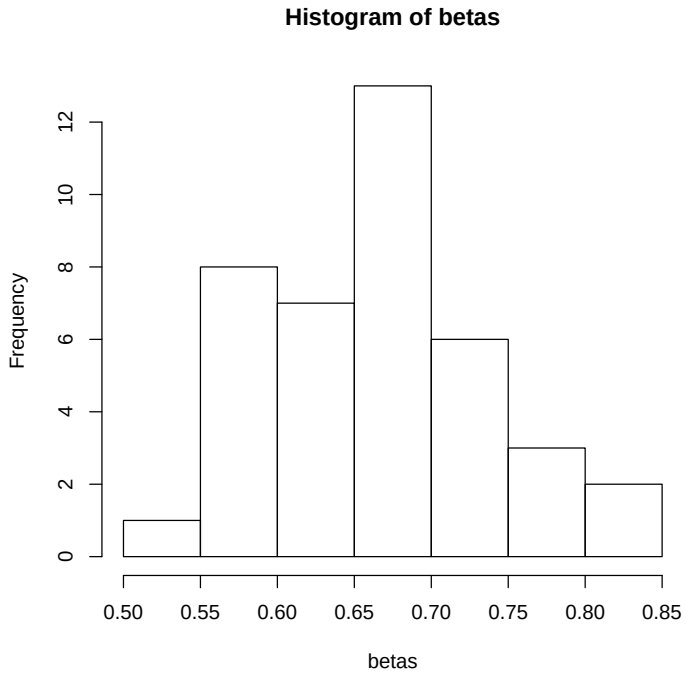

So inspite of the non-naturalness of the resulting figures, we can still run a bootstrap. But we aren't impressed with the simulation since it really looks different than the truth.

```
> number.bins <- 20
> bins <- trunc(number.bins * (1:dim(both)[1] - 1)/dim(both)[1])
> repeats <- 40
> betas <- rep(NA, repeats)
> for (i in 1:repeats) {
+     indexes <- block_sample(bins)
+     bootstrap <- both[indexes, ]
```

```
+     simple.model <- lm(bootstrap$Obama.Close ~ bootstrap$SP.500.Close)
+     betas[i] <- simple.model$coefficients[2]
+ }
```

This lowers the t-statistics alot, down to 9.39979507339954, but still big enough to be impressive. Its histogram also makes it look plausable:

```
> hist(betas)
```

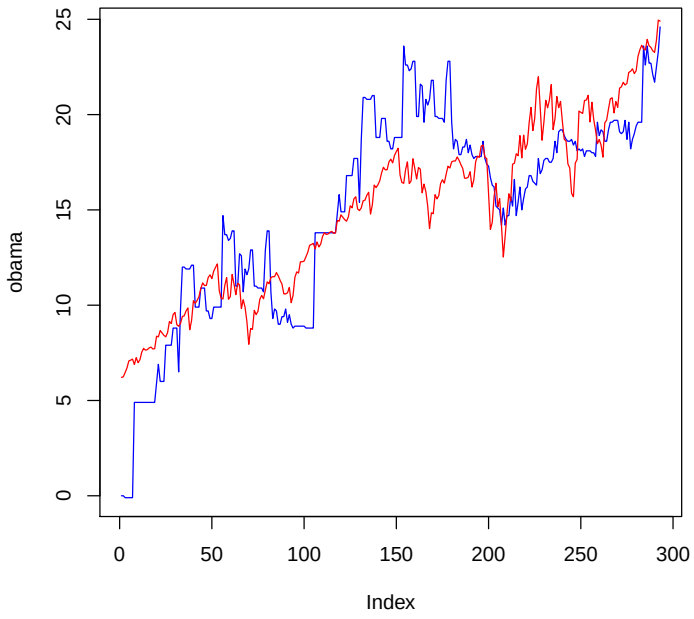



## Time series bootstrap on differences

Let's chop this up into 20 segments (about one month each) and bootstrap these months. But we will make it so that the months link to each other. We will start out by looking at what such a series looks like:

```
> number.bins <- 3
> bins <- trunc(number.bins * (1:dim(both)[1] - 1)/dim(both)[1])
> indexes <- block_sample(bins)
> bootstrap <- both[indexes, ]
> bootstrap$SP.diffs[is.na(bootstrap$SP.diffs)] = 0
> bootstrap$Obama.diffs[is.na(bootstrap$Obama.diffs)] = 0
> SP = cumsum(bootstrap$SP.diffs)
```

```
> obama = cumsum(bootstrap$Obama.diffs)
> boot.model <- lm(obama ~ SP)
> plot(obama, col = "blue", type = "l")
> alpha <- boot.model$coefficients[1]
> beta <- boot.model$coefficients[2]
> lines(alpha + beta * SP, col = "red")
```




Now to see the actual bootstrap:

```
> number.bins <- 20
> bins <- trunc(number.bins * (1:dim(both)[1] - 1)/dim(both)[1])
> repeats <- 400
> betas <- rep(NA, repeats)
> for (i in 1:repeats) {
+     indexes <- block_sample(bins)
+     bootstrap <- both[indexes, ]
+     bootstrap$SP.diffs[is.na(bootstrap$SP.diffs)] = 0
+     bootstrap$Obama.diffs[is.na(bootstrap$Obama.diffs)] = 0
+     SP = cumsum(bootstrap$SP.diffs)
+     obama = cumsum(bootstrap$Obama.diffs)
+     simple.model <- lm(obama ~ SP)
+     betas[i] <- simple.model$coefficients[2]
+ }
```

This lowers the t-statistics alot, down to 1.08837662330499, but still big enough to be impressive. Its histogram also makes it look plausable:

```
> hist(betas)
```