HW-Section3

Rob McCulloch

July 24, 2020

1 Homework for Section 3

1.1 SLR Model

Suppose we are modeling house price as depending on house size. Price is measured in thousands of dollars and size is measured in thousands of square feet.

Suppose our model is:

$$P = 20 + 50s + \epsilon, \epsilon \sim N(0, 15^2).$$

That is, suppose that somehow we *know* the parameters:
$\beta_0 = 20$,
$\beta_1 = 50$, and
$\sigma = 15$.

(a)

Given you know that a house has size s = 1.6, give a 95% predictive interval for the price of the house.

(b)

Given you know that a house has size s = 2.2, give a 95% predictive interval for the price.

(c)

In our model the slope is 50. What are the units of this number?

(d)

What are the units of the intercept 20?

(e)

What are the units of the standard deviation 15?

(f)

Suppose we change the units of price to dollars and size to square feet.
What would the values and units of the intercept, slope, and error standard deviation?

(g)

If we plug s = 1.6 into our model equation (with the original units), P is a constant plus the normal random variable ϵ.

Given s = 1.6, what is the distribution of P ?

Solution

(a)

The point prediction is $P_f = 20 + 50 * 1.6 = 100$ (P_f is the "future P").

The prediction interval is $(100 \pm 2 * 15) = (70, 130)$.

(b)

The point prediction is $P_f = 20 + 50 * 2.2 = 130$ The prediction interval is $[130 \pm 2 * 15] = [100, 160]$.

(c)

\$1,000 / 1,000 Sq. Feet = \$/Sq. Feet

(d)

Same as the units of the response P, thousands of dollars.

(e)

\$1,000 (same as P)

(f)

Intercept: 20,000 \$ Slope: 50 \$/Sq. Feet error standard deviation: 15,000 \$

(g)

When $s = 1.6$ the mean of house prices is $20 + 50 * 1.6 = 100$. The error standard deviation is the same, 15. Therefore

$$P \mid S = 1.6 \sim \mathrm{N}(100, 15^2)$$

1.2 The Shock Absorber Data

The data comes from a company which supplies a major automobile manufacturer with shock absorbers. An important characteristic is the "force transferred through the shock absorber when the shank is forced out of the cylinder". If you don't know what that really means, don't worry, neither do I.

What we do need to understand is that the manufacturer only considers the shock to be an acceptable part if the force measurement is between 485 and 585.

The shock manufacturer and the auto manufacturer are arguing over the following issue. Before the shock is finally shipped, it is filled with gas. After it is filled with gas, it becomes very difficult to measure the force characteristic we are interested in. The shock manufacturers would like to make the measurement before the shock is filled with gas. The auto maker is concerned that there may be a difference in the force before and after the shock is filled with gas and so would like to make the measurement after it is filled.

The shock maker claims that there is little difference between the before and after measurement so that the before measurement can be used.

To investigate this we have the before (column 1, reboundb) and the after (column 2, rebounda) measurements on 35 shocks (in shock.csv).

Get the shock data (shock.csv) from the webpage.

(a)

Plot reboundb vs. rebounda.

Does this look like the kind of data the simple linear regression model is designed to capture?

```
R:
sdat = read.csv("http://www.rob-mcculloch.org/data/shock.csv")
plot(sdat)
```

(b)

Run the regression of y=rebounda on x=reboundb.

What is the estimate of the true slope?

(c)

Given reboundb = 535, give the plug-in predictive interval for rebounda.

(d)

Give the 95% confidence interval for β_1.

(e)

Give the 95% confidence interval for β_0.

(f)

Test the null hypothesis (level .05) that $\beta_0 = 0$.

(g)

Test the null hypothesis (level .05) that $\beta_1 = 0$.

(h)

Test the null hypothesis (level .05) that $\beta_1 = 1$.

Why is this an interesting hypothesis to test?

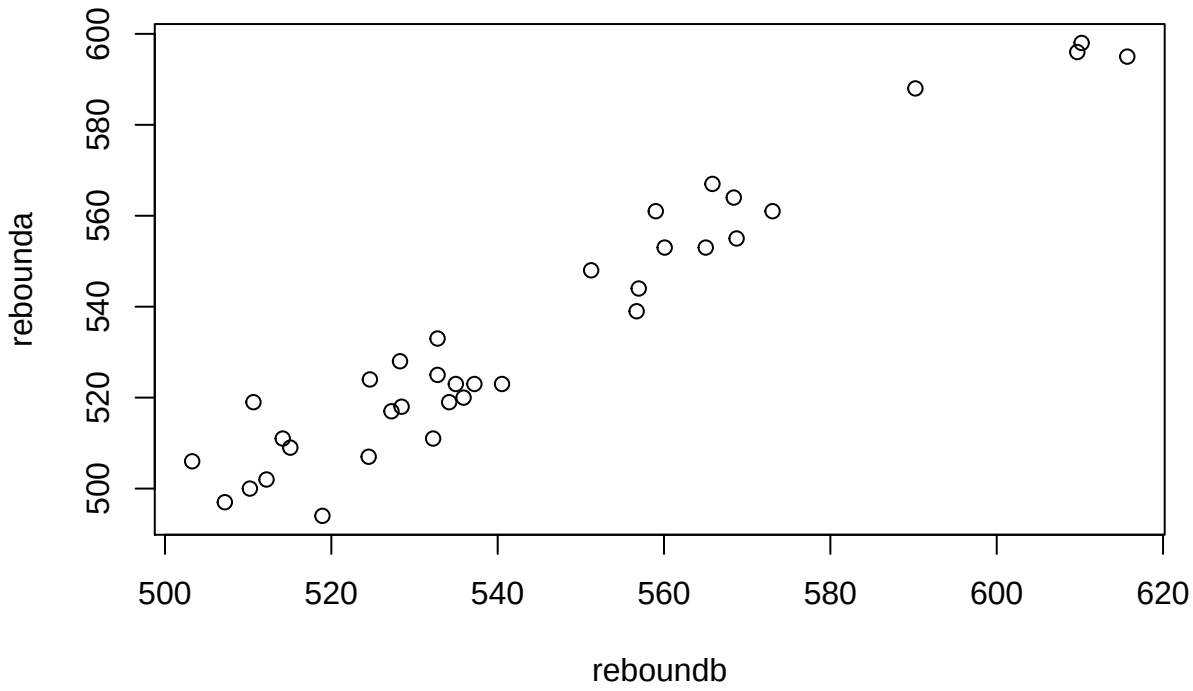
(i)

Is it ok to use the before measurement as a proxy for the after measurement? What does the simple linear regression model tell us about this?

Solution

(a)

```
sdat = read.csv("http://www.rob-mcculloch.org/data/shock.csv")
plot(sdat)
```



Yes, really looks like line +- error !!

(b)

```
lmsa = lm(rebounda~reboundb,sdat)
summary(lmsa)
```

```
##
## Call:
## lm(formula = rebounda ~ reboundb, data = sdat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -16.931  -4.246  -1.692   6.250  15.940
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.2259    23.8852   0.763    0.451
## reboundb      0.9495     0.0438  21.675   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.67 on 33 degrees of freedom
## Multiple R-squared:  0.9344, Adjusted R-squared:  0.9324
## F-statistic: 469.8 on 1 and 33 DF,  p-value: < 2.2e-16
```

$\hat{\beta}_1 = 0.9495$.

(c)

```
18.2259 + 0.9495*535 + 7.67*2*c(-1,1)
```

```
## [1] 510.8684 541.5484
```

(d)

```
 0.9495 + 2*0.0438*c(-1,1)
```

```
## [1] 0.8619 1.0371
```

(e)

```
18.2259 + 2* 23.8852*c(-1,1)
```

```
## [1] -29.5445  65.9963
```

(f)

pval = .451, fail to reject

(g)

pval = 0, reject

(h)

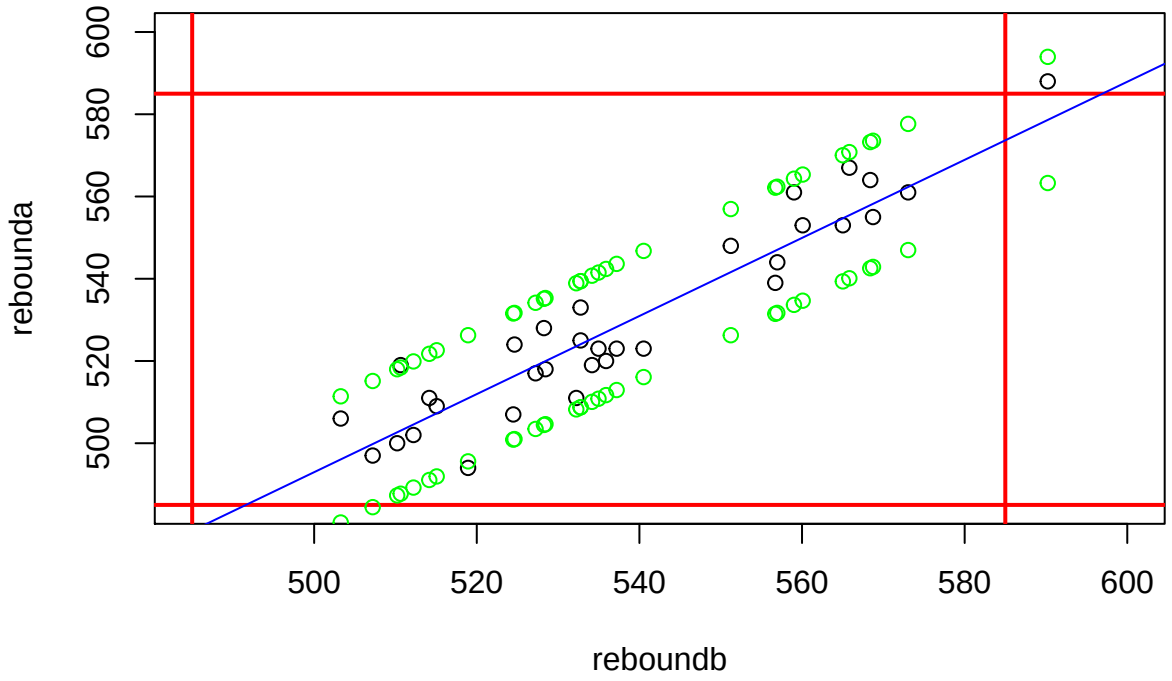
```
t = ( 0.9495-1)/0.0438
t
```

```
## [1] -1.152968
```

fail to reject. slope=1 and intercept=0 would be consistent with using the before as a proxy for the after.

(i)

```
plot(sdat,xlim=c(485,600),ylim=c(485,600))
abline(h=485,col="red",lwd=2)
abline(h=585,col="red",lwd=2)
abline(v=485,col="red",lwd=2)
abline(v=585,col="red",lwd=2)
abline(lmsa$coef,col="blue")
points(sdat$reboundb,lmsa$fitted + 2*7.67,col="green")
points(sdat$reboundb,lmsa$fitted - 2*7.67,col="green")
```



Red is tolerance limits of (485,585).
Green is plug-in predictive intervals.
Blue is fitted line.

Short answer is yes.
If before is in, you can be pretty sure after is in.
Might want to double check if before is close to one of the limits.

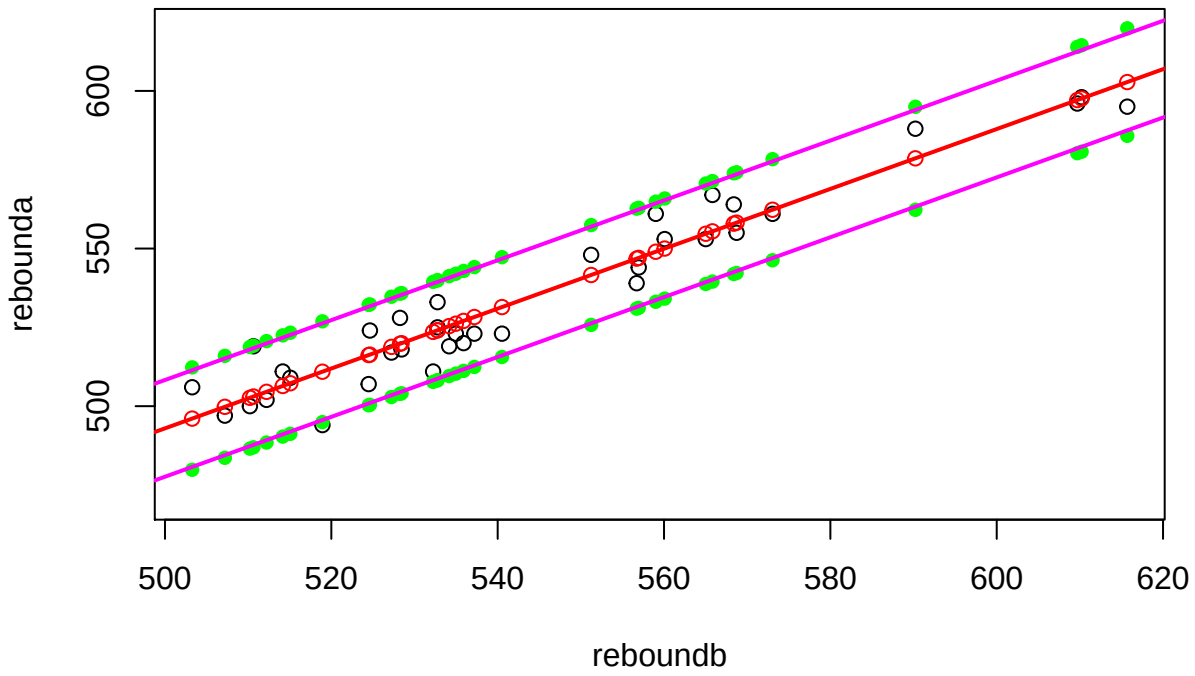
1.3 Predictive Interval for the Shock Data

Let's compare the plug in predictive interval with the "correct" predictive interval (the one the accounts for our estimation error) for the shocks data.

Is there enough information in the data to make the plug-in interval similar to the predictive interval??

Here is the R code to get and plot the predictive and plug-in intervals.

```
sd = read.csv("http://www.rob-mcculloch.org/data/shock.csv")
lms = lm(reboundaˆreboundb,sd)

#note: try > ?predict.lm

predint = predict(lms,sd,interval="prediction")

plot(sd,ylim=c(470,620))
points(sd$reboundb,predint[,"fit"],col="red") #note predint is a matrix, not a data.frame
points(sd$reboundb,predint[,"lwr"],col="green",pch=16)
points(sd$reboundb,predint[,"upr"],col="green",pch=16)

sigmahat = summary(lms)$sigma #estimate of sigma
ab = coef(lms) #estimates of intercept and slope
abline(ab[1]+2*sigmahat,ab[2],col="magenta",lwd=2)
abline(ab[1]-2*sigmahat,ab[2],col="magenta",lwd=2)
abline(ab[1],ab[2],col="red",lwd=2)
```



Solution

Wow! The plug-in interval coincides very closely with the predictive interval, so YES!!

1.4 Beta for Fidelity Funds

Get the data in the file fidrets.csv.

The data is monthly returns on: sp500: the s&p 500.
FidInc: a Fidelity income fund.
FidVal: a Fidelity "value" fund.
FidTech: a Fidelity Tech fund.

From the names, we might expect the value fund to be riskier than the income fund and the tech fund to be riskier than the value fund.

(a)

For each of the three funds plot sp500 vs. fund return.
Is linear regression a good way to think about the relationship between
the market returns and the fund returns?

So that you can compare the different funds, make sure each plot is on the same scale.

(b)

For each of the three funds compute the 95% confidence interval for the slope.

What do these intevals test us about the risk of the funds?

(c)

This is not well motivated in the example, but just as an excercise for each of the three funds test $\beta_1 = 0$ and $\beta_1 = 1$ (the slope is 0 and the slope is 1).

Solution

(a)

Let's read in the data:

```
ff = read.csv("http://www.rob-mcculloch.org/data/fidrets.csv")
print(names(ff))
```

```
## [1] "sp500"   "FidInc"  "FidVal"  "FidTech"
```

```
print(dim(ff))
```

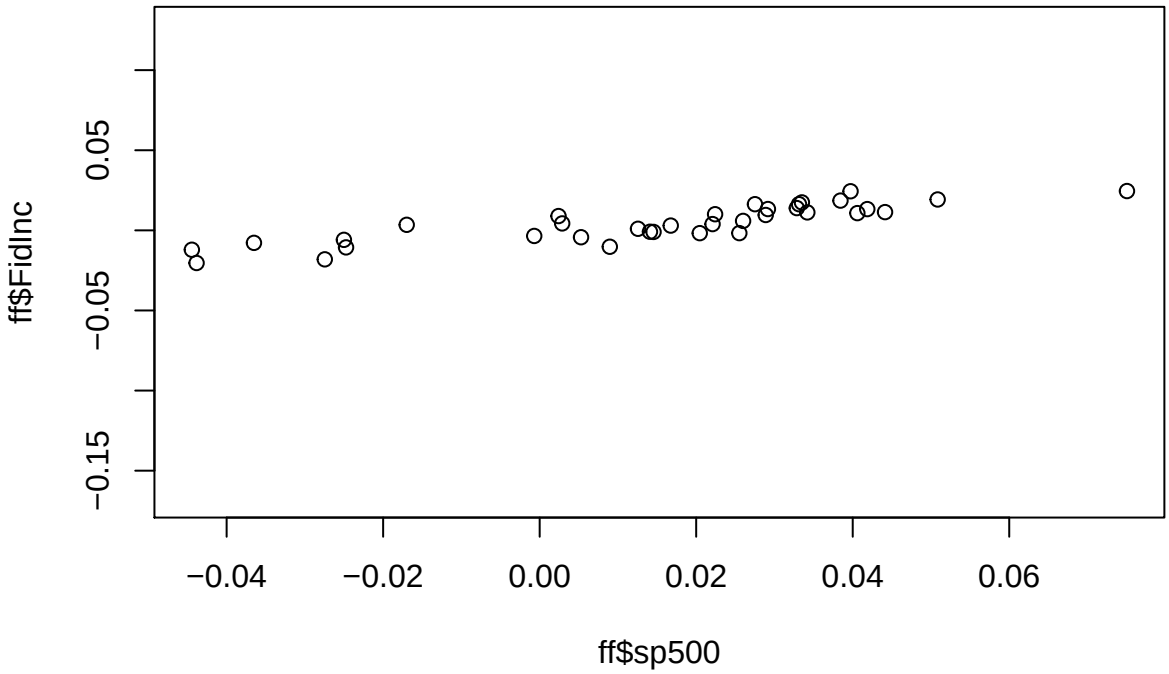
```
## [1] 35  4
```

```
head(ff)
```

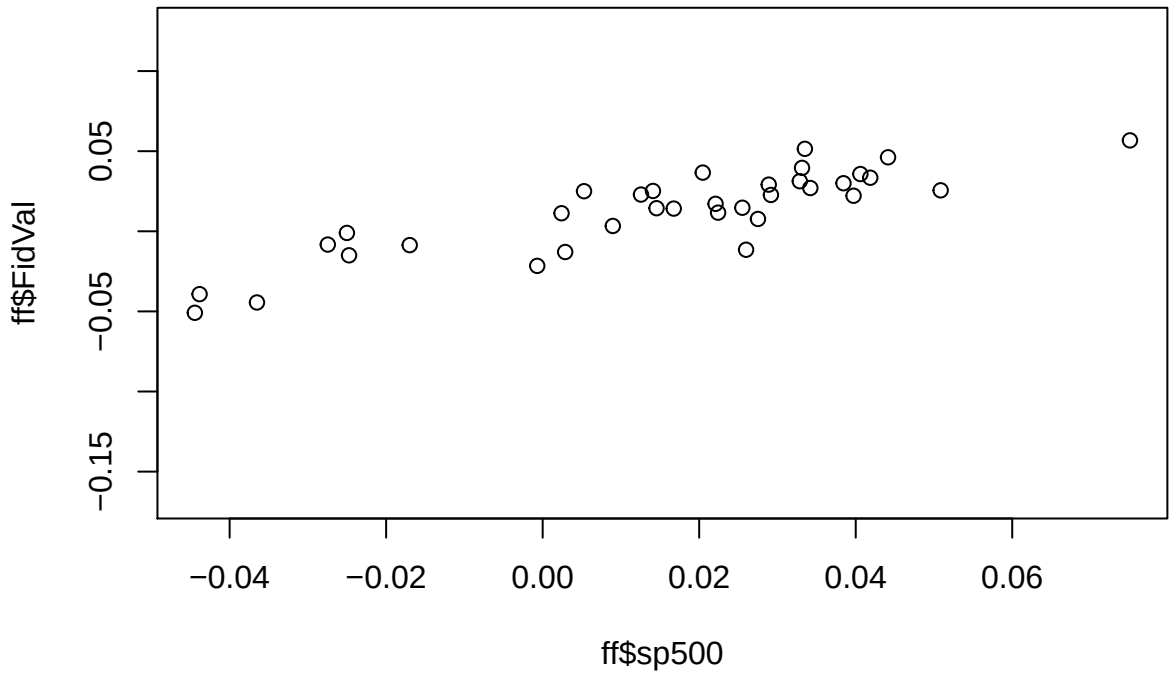
```
##         sp500       FidInc       FidVal      FidTech
## 1  0.03350438  0.017451720  0.051467541  0.050443459
## 2 -0.02745619 -0.018095238 -0.008285643  0.028232190
## 3 -0.04384097 -0.020368574 -0.039230674 -0.033872209
## 4  0.01256281  0.000980198  0.022921557 -0.019657371
## 5  0.01674628  0.002977280  0.014164306  0.001628289
## 6 -0.02501533 -0.005927022 -0.001015744 -0.084663240
```

And not let's plot all three:

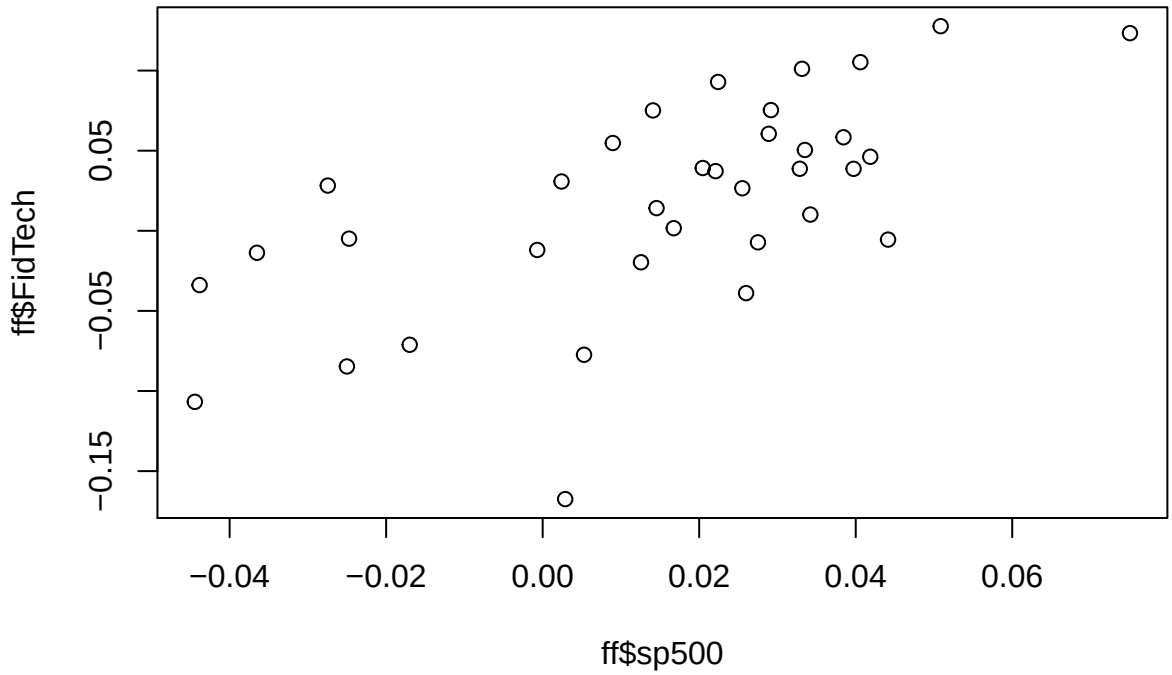
```
# first get limits for the plots, we want them on the same scale
xlim = range(ff$sp500)
ylim = range(ff[,2:4])

# three plots in same figure
plot(ff$sp500,ff$FidInc,xlim=xlim,ylim=ylim)
```




```
plot(ff$sp500,ff$FidVal,xlim=xlim,ylim=ylim)
```




```
plot(ff$sp500,ff$FidTech,xlim=xlim,ylim=ylim)
```



Very dramatic how different relationship to the market is for each of our three funds.!!

(b)

```
lmFidInc = lm(FidInc~sp500,ff)
summary(lmFidInc)
```

```
##
## Call:
## lm(formula = FidInc ~ sp500, data = ff)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0127402 -0.0034358 -0.0009288  0.0041676  0.0109666
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0006729  0.0011224  -0.599    0.553
## sp500        0.3550981  0.0356225   9.968 1.75e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005842 on 33 degrees of freedom
## Multiple R-squared:  0.7507, Adjusted R-squared:  0.7431
## F-statistic: 99.37 on 1 and 33 DF,  p-value: 1.753e-11
```

Confidence Interval for Income beta is:

```
.355 + 2*.0356*c(-1,1)
```

```
## [1] 0.2838 0.4262
```

Regression for `FidVal`:

```
lmFidVal = lm(FidVal~sp500,ff)
summary(lmFidVal)
```

```
##
## Call:
## lm(formula = FidVal ~ sp500, data = ff)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.033008 -0.006692 -0.000823  0.009432  0.023928
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.0005589  0.0025190   0.222    0.826
## sp500       0.8052956  0.0799444  10.073 1.35e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01311 on 33 degrees of freedom
## Multiple R-squared:  0.7546, Adjusted R-squared:  0.7472
## F-statistic: 101.5 on 1 and 33 DF,  p-value: 1.348e-11
```

Confidence Interval for Value beta is:

```
.8 + 2*.08*c(-1,1)
```

```
## [1] 0.64 0.96
```

```
lmFidTech = lm(FidTech~sp500,ff)
summary(lmFidTech)
```

```
##
## Call:
## lm(formula = FidTech ~ sp500, data = ff)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.166180 -0.033587  0.005578  0.037878  0.075195
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.005604   0.009821  -0.571    0.572
## sp500        1.506359   0.311696   4.833 3.02e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05112 on 33 degrees of freedom
## Multiple R-squared:  0.4144, Adjusted R-squared:  0.3967
## F-statistic: 23.36 on 1 and 33 DF,  p-value: 3.017e-05
```

Confidence Interval for Tech beta is:

```
1.5 + 2*.311*c(-1,1)
```

```
## [1] 0.878 2.122
```

The intervals are big as a practical matter, but there are small enough to strongly suggest that the beta's are increasing, suggesting FidInc is less risky than FidVal and FidVal is less risky that FidTech, which makes sense.

(c)

For testing slope=0 all the p-values are tiny => reject.

For testing =1:

```
> #Income t
> (0.3550981-1)/0.0356225
[1] -18.10378
> #reject
>
> #Value t
> (0.8052956-1)/0.0799444
[1] -2.435498
> #reject
>
> #Tech t
> (1.506359-1)/0.311696
[1] 1.624528
> #fail to reject at usual levels.
> 2*pnorm(-1.624528)
[1] 0.1042632
> #p-value is about .1
```

1.5 Correlation in Simple Linear Regression, the shock absorber data

Let's use correlation to think about the regression we ran for the shock absorber example.

```
sdat = read.csv("http://www.rob-mcculloch.org/data/shock.csv")
cor(sdat)
```

```
##           reboundb  rebounda
## reboundb 1.0000000 0.9666268
## rebounda 0.9666268 1.0000000
```

(a)

We see that the correlation betweeen `reboundb` and `rebounda` is .966.

What does this tell us about the relationship?

(b)

Get the fitted values $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_i$ for y=rebounda x=reboundb.

What is the correlation between x and the fitted values ?

Plot `x` vs. the fitted values.

Note that in `R` we could compute the fits directly from the formula:

```
lmsh = lm(rebounda~reboundb,sdat)
bhat = lmsh$coef
bhat
```

```
## (Intercept)    reboundb
##  18.2258676   0.9494632
```

```
fits = bhat[1] + bhat[2] * sdat$reboundb
```

Or, we can get the fits directly from the regression results:

```
names(lmsh)
```

```
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
```

```
fits1 = lmsh$fitted.values
summary(fits-fits1) ## check that they are the same
```

```
##       Min.    1st Qu.     Median       Mean    3rd Qu.       Max.
## -1.137e-13 -5.684e-14  0.000e+00 -3.086e-14  0.000e+00  1.137e-13
```

The fits and resids play an important role in regression modeling, so they are included in the results in `R`.

(c)

Plot the fitted values vs y and x vs y.
How do these plots compare?

What is the correlation between the fitted values and y?
How does this compare to the correlation between x and y?

(d)

Square the correlation between x and y.
How does this compare to the "Multiple R-squared" in the regression summary?

(e)

What is the correlation between x and the residuals? What is the correlation between the fits and the residuals?

Solution

(a)

The correlation is very close to one. In our application, this would make us optimistic that the relationship is linear and strong enough for the before measurement to be a good proxy. But only the regression anlysis gives us a real answer!!

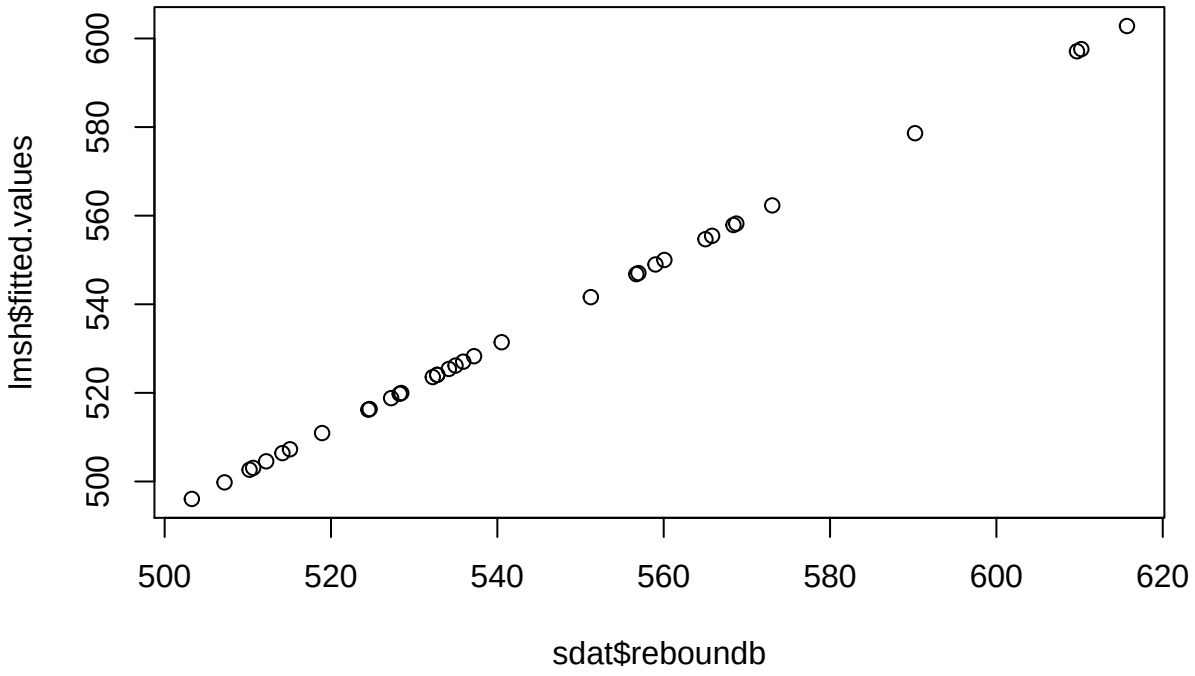
(b)

```
cor(sdat$reboundb,fits)
```

```
## [1] 1
```

Because the fitted values are an exact linear function of `x`, the correlation between the fits and `x` is 1!!!

```
plot(sdat$reboundb,lmsh$fitted.values)
```



(c)

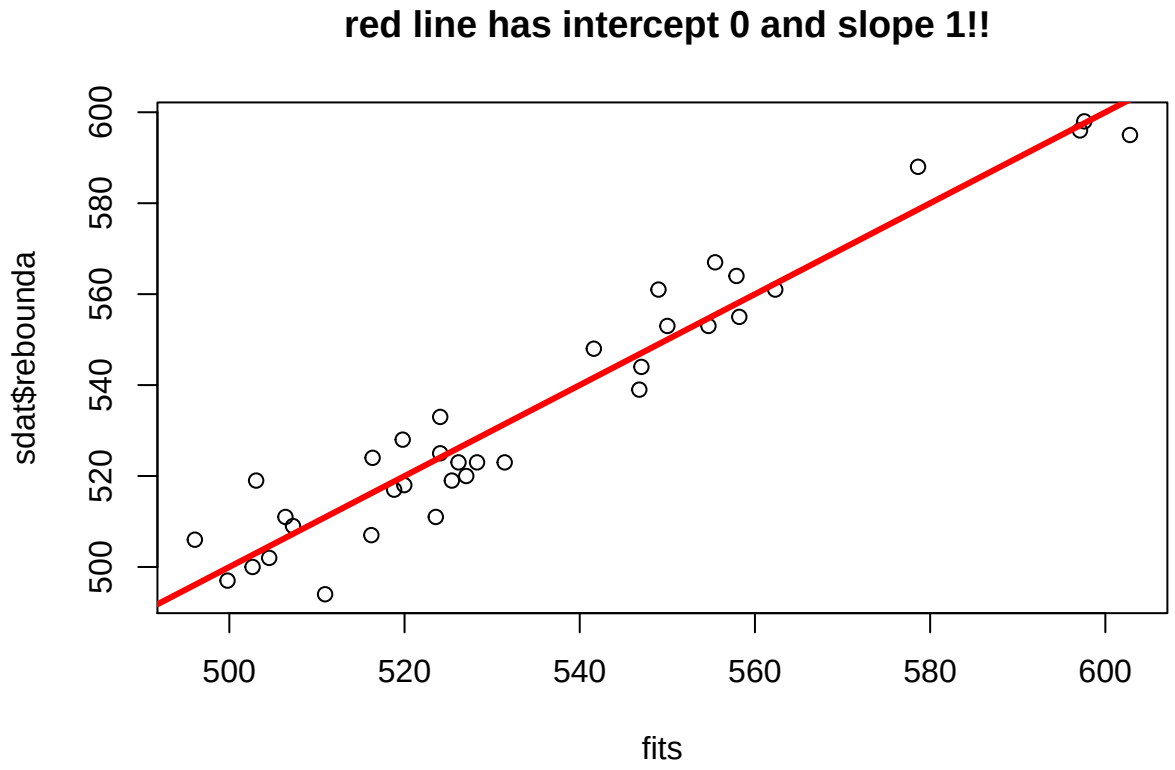
```
cor(fits,sdat$rebounda)
```

```
## [1] 0.9666268
```

$cor(y, \hat{y}) = cor(x, y)$ in simple linear regression!!

This is because $\hat{y}$ is an exact linear function of x, the fits are just x rescaled to look like y!!

```
plot(fits,sdat$rebounda)
abline(0,1,col="red",lwd=3)  #line with intercept 0, and slope 1!!
title(main=" red line has intercept 0 and slope 1!!")
```



The plot of fits vs y=rebounda *looks* just like the plot of x vs y, but note the the scale on the x axis is different.
With one x, the fits are just x shifted (intercept) and scaled (slope) to look like y.

(d)

They are the same!!
In simple linear regression, R-squared is r squared!!

(e)

```
print(cor(lmsh$residuals,sdat$reboundb))
```

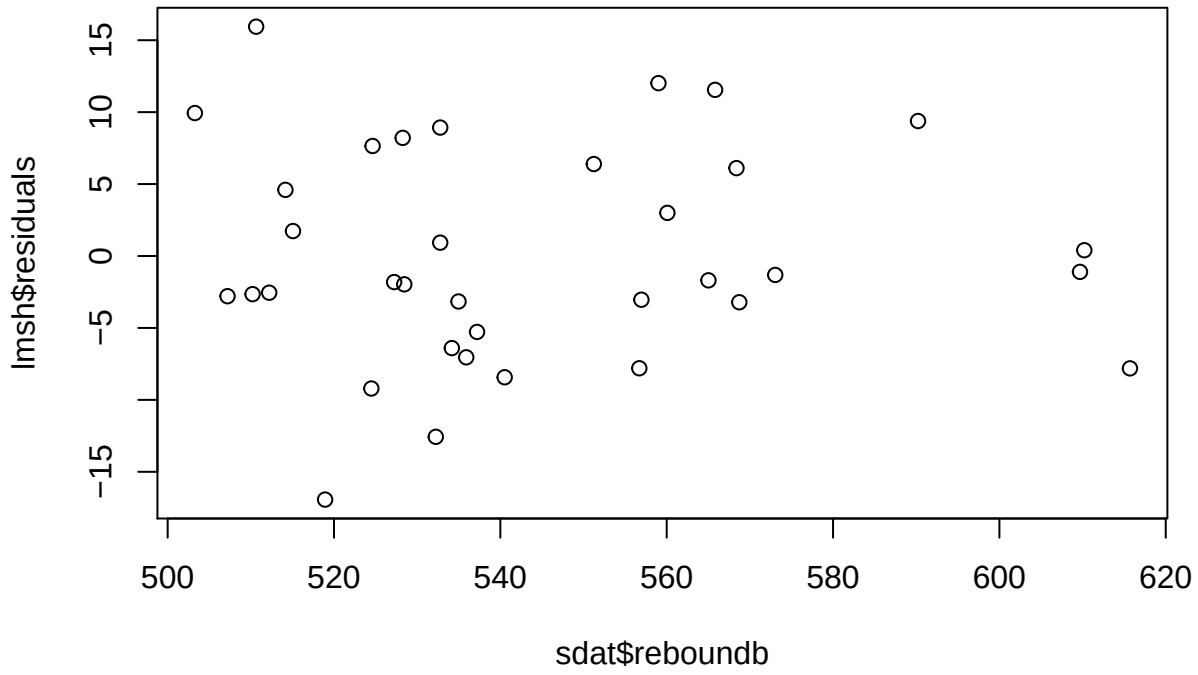
```
## [1] -7.511949e-16
```

```
print(cor(lmsh$residuals,fits))
```

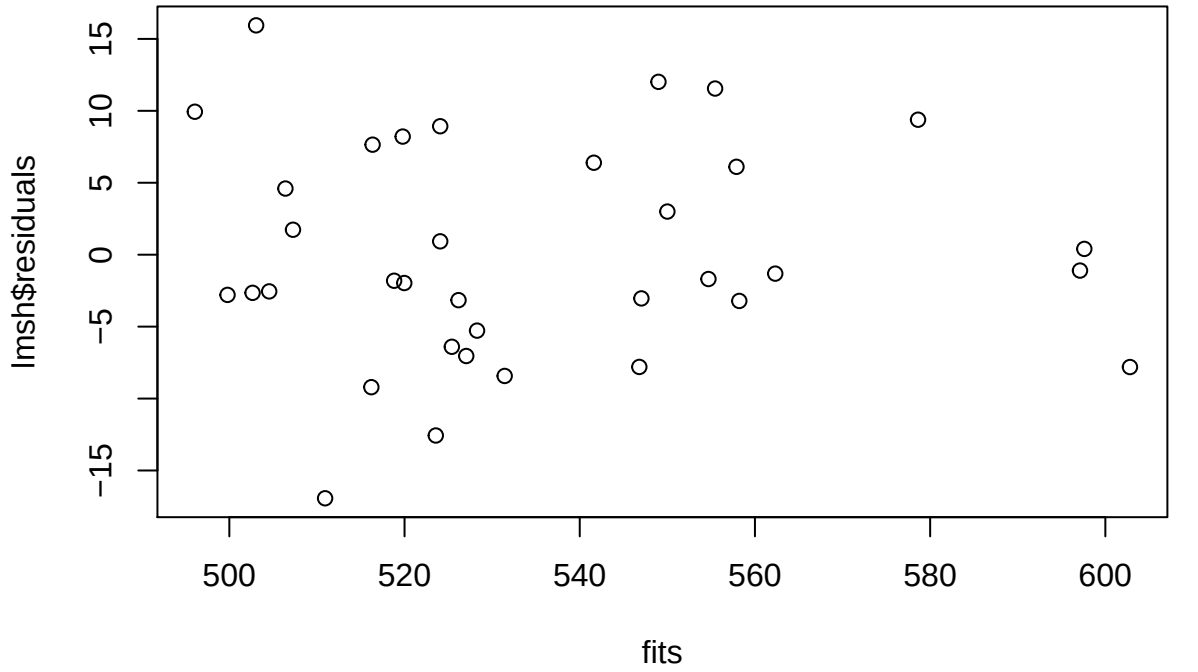
```
## [1] -9.576525e-16
```

They are both 0!!

```
plot(sdat$reboundb,lmsh$residuals)
```




```
plot(fits, lmsh$residuals)
```



No linear relationship! The resids are the part of y the does not look like a linear function of x!!

Again, these look the same because the fits are just x rescaled.

1.6 Correlation and the Countries Returns

Get the returns data for various countries (conret.csv).
Based on the correlations, what countries move together?

Solution

```
cdat = read.csv("http://www.rob-mcculloch.org/data/conret.csv")
```

```
cor(cdat[,c(4,6,8,9)])
```

```
##           belgium   denmark    france   germany
## belgium 1.0000000 0.5342116 0.7335749 0.6909786
## denmark 0.5342116 1.0000000 0.4877015 0.6166755
## france  0.7335749 0.4877015 1.0000000 0.7090513
## germany 0.6909786 0.6166755 0.7090513 1.0000000
```

```
cor(cdat[,c('canada','usa')])
```

```
##           canada       usa
## canada 1.0000000 0.6511531
## usa    0.6511531 1.0000000
```

Makes sense!

1.7 Portolio of Fidelity Returns

Use the data fidret.csv.

Using the sample quantities as estimates for the true means, variances, and covariance what is your plug in estimate of the mean and standard deviation of a portfolio which puts 25% into the value fund and 75% into the tech fund?

Solution

```
fr = read.csv("http://www.rob-mcculloch.org/data/fidrets.csv")
names(fr)
```

```
## [1] "sp500"   "FidInc"  "FidVal"  "FidTech"
```

```
muhatVal = mean(fr$FidVal); sighatVal = sd(fr$FidVal)
muhatTec = mean(fr$FidTech); sighatTec = sd(fr$FidTech)
cVT = cov(fr$FidVal,fr$FidTech)
cat("two means, two sds, and cov:",muhatVal,muhatTec,sighatVal,sighatTec,cVT)
```

```
## two means, two sds, and cov: 0.01261904 0.01695538 0.02607556 0.06581641 0.001074193
```

```
EPhat = .25*muhatVal + .75*muhatTec
VPhat = .25^2 * sighatVal^2 + .75^2 * sighatTec^2 + 2*.25*.75*cVT
cat("estimated mean, var, and sd of Porftolio:",EPhat,VPhat,sqrt(VPhat),"\n")
```

```
## estimated mean, var, and sd of Porftolio: 0.0158713 0.002881956 0.05368385
```