A closer look at "How economists get tripped up by statistics"

Laurie Samuels

August 9, 2012

1 Introduction

Chris Fonnesbeck sent Jeffrey a link to an interesting blog post, "How economists get tripped up by statistics" (`http://blogs.reuters.com/felix-salmon/2012/07/10/how-economists-get-tripped-up-by-statistics/`). The original article referenced in the blog post is `http://emresoyer.com/Publications_files/Soyer%20%26%20Hogarth_2012.pdf`; it has been published as Soyer, E., & Hogarth, R. M. (2012). The illusion of predictability: How regression statistics mislead experts. International Journal of Forecasting, 28(3), 695-711. doi:10.1016/j.ijforecast.2012.02.002.

Jeffrey thought it would be interesting to reproduce the scenario presented in the blog (Conditions 1, 3, and 5 in the original paper), and that's what I've tried to do here. I also thought it would be a good chance to practice using knitr (`http://yihui.name/knitr/`); it seems like a big improvement over Sweave, and I hope this example will help other people use it too.

Soyer and Hogarth asked their subjects four questions; here we'll look only at the one presented in Salmon's blog post. Salmon slightly misstated the question in the post, however, so we'll use Soyer and Hogarth's original wording: "What would be the minimum value of X that an individual would need to make sure that s/he obtains a positive outcome ($Y > 0$) with 95% probability?" We'll try to answer the question first using just the scatterplot, then using just the regression output, and then finally using both the scatterplot and the regression output.

2 Answering the question using only the scatterplot

First of all, let's try to make a scatterplot like the one Soyer and Hogarth presented:

```
n     <- 1000
xmean <- 50.72
ymean <- 51.11
xsd   <- 28.12
ysd   <- 40.78
```

```
R2    <- 0.5
rxy   <- sqrt(R2)
covxy <- rxy * xsd * ysd
```

Soyer and Hogarth used a sample of size 1000; see Table 1 for details.

Variable	Mean	SD
x	50.72	28.12
y	51.11	40.78

Table 1: Sample characteristics

One way to recreate their scatterplot might be to generate a bivariate normal random sample with the same means and covariance structure as the Soyer and Hogarth data. We have the means and standard deviations of X and Y, but what is $cov(X, Y)$? Because they used simple linear regression, we can get the sample correlation by taking the square root of the R^2 from their regression model; and we can get the covariance from this correlation and the two standard deviations. Their value for R^2 was 0.5, so our value for r_{xy} is 0.7071 and our value for $cov(x, y)$ is 810.8631. If we want to generate a bivariate normal random sample with these characteristics, we can do so as follows:

```
covMat <- matrix(c(xsd^2, covxy, covxy, ysd^2), ncol= 2)
s1 <- mvrnorm(n= n, mu= c(xmean, ymean), Sigma= covMat, empirical= TRUE)
s1 <- data.frame(s1)
names(s1) <- c("x", "y")
```

Another way to generate a sample similar to Soyer and Hogarth's might be to consider the x's to be fixed, rather than random, and to generate the Y's using the RMSE and approximate coefficients from the regression model, as follows:

```
x2 <- 100 * runif(1000)
e2 <- rnorm(1000, mean= 0, sd= 29)
y2 <- 1 * x2 + e2
s2 <- data.frame(x= x2, y= y2)
```

```
# prepare a summary table
r1 <- c(xmean, xsd, rxy, mean(s1$x), sd(s1$x), cor(s1)[1,2], mean(s2$x),
    sd(s2$x), cor(s2)[1,2])
r2 <- c(ymean, ysd, NA, mean(s1$y), sd(s1$y), NA, mean(s2$y), sd(s2$y), NA)
compTable <- data.frame(rbind(r1, r2))
compTableRounded <- round(compTable, 2)
```

Let's compare the summary statistics from the two methods:

Variable	Soyer and Hogarth			Bivariate Normal			Fixed x		
	mean	sd	cor(x,y)	mean	sd	cor(x,y)	mean	sd	cor(x,y)
x	50.72	28.12	0.71	50.72	28.12	0.71	49.47	28.74	0.71
y	51.11	40.78		51.11	40.78		49.99	40.74	

Table 2: Comparison of summary statistics from two sample-generation techniques (and Soyer and Hogarth's data)

Figure 1 and Figure 2 are scatterplots from the two methods.

Figure 2 more closely resembles the graph from Soyer and Hogarth's paper, so we'll use that one from here on.

```r
# scatterplot for BVN sample
with(s1, plot(x, y, xlim= c(0,100), xaxp= c(0, 100, 20), ylim= c(-150, 250)))
m1 <- lm(y ~ x, data= s1)
abline(h= 0, col= "grey50")
abline(m1)
summM1 <- summary(m1)
m1Sigma <- summM1$sigma
```

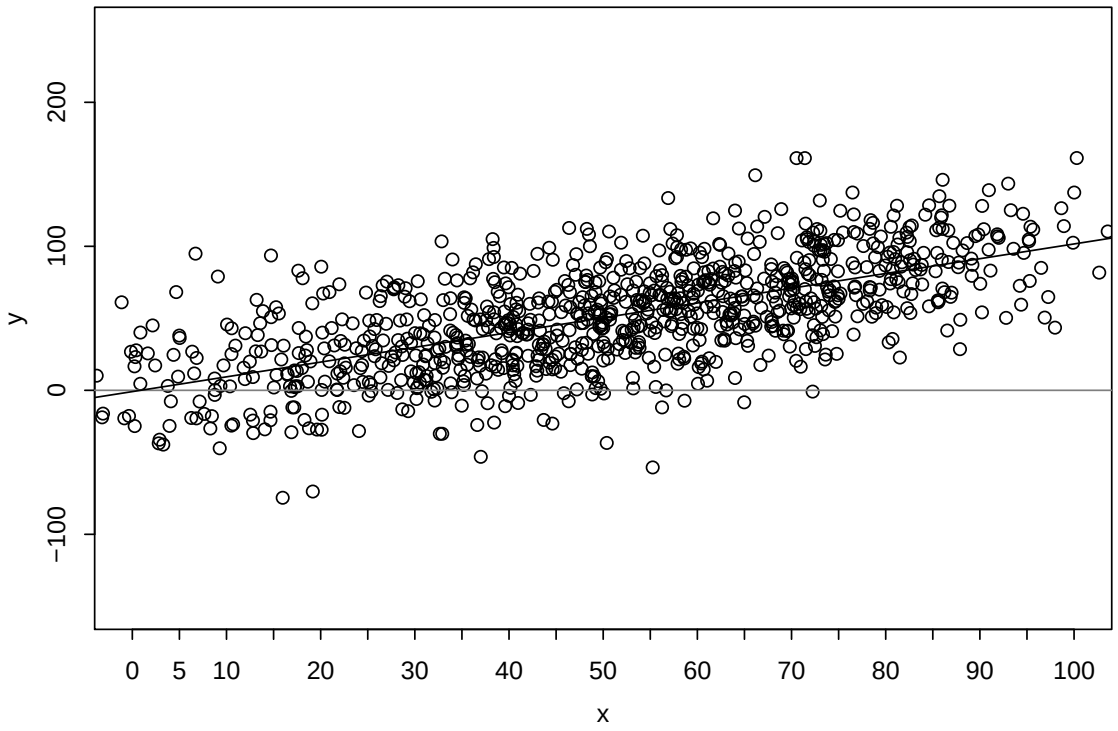

Figure 1: Bivariate normal data. The standard error of the regression ($\hat{\sigma}_e$) is 28.8503 (compare to Soyer & Hogarth's value of 29).

```
# scatterplot for sample generated from "fixed" x's
with(s2, plot(x, y, xlim= c(0,100), xaxp= c(0, 100, 20), ylim= c(-150, 250)))
m2 <- lm(y ~ x, data= s2)
abline(h= 0, col= "grey50")
abline(m2)
summM2 <- summary(m2)
m2Sigma <- summM2$sigma
```

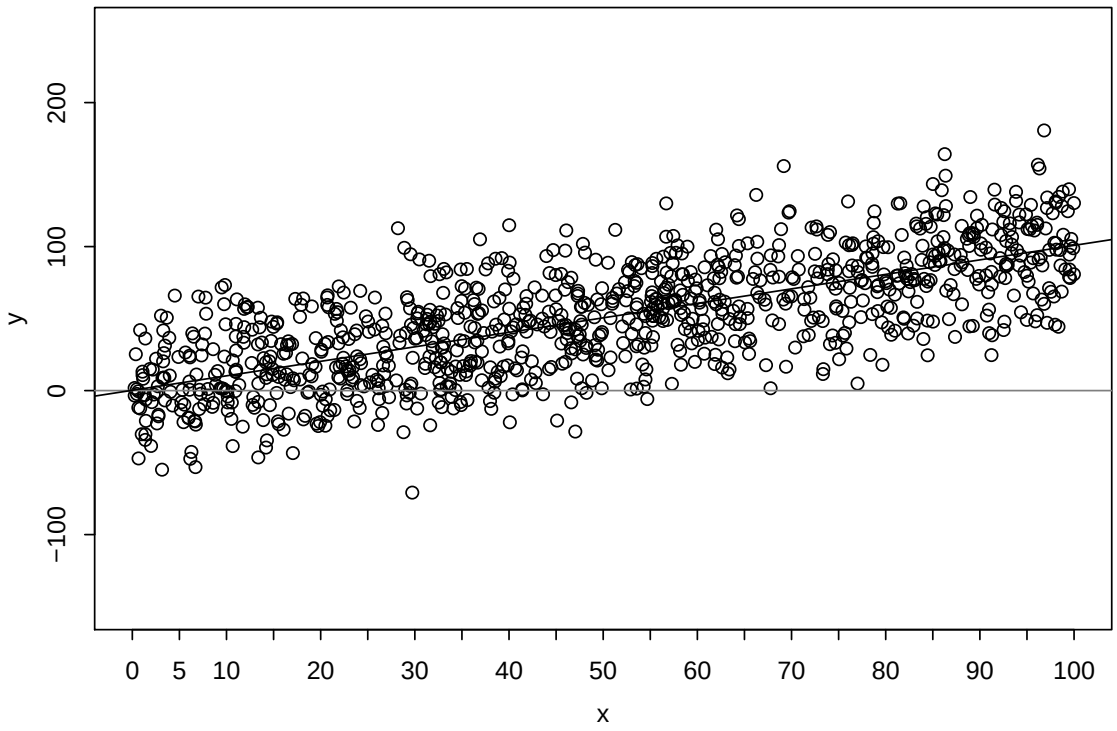

Figure 2: Data generated from fixed x's. The standard error of the regression ($\hat{\sigma}_e$) is 28.6968 (compare to Soyer & Hogarth's value of 29).

Soyer and Hogarth asked people, "What would be the minimum value of X that an individual would need to make sure that s/he obtains a positive outcome ($Y > 0$) with 95% probability?" We can answer this question using only Figure 2 if we are comfortable making a few assumptions:

1. The observations are independent.
2. In the population, the mean value of Y really is a linear function of X.
3. The population variance of Y is the same for all values of X; or, $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, where ϵ is distributed with mean 0 and constant variance σ^2. I don't think normality is required for us to answer this question visually, but I'm not totally sure about that.

Based on Figure 2, these assumptions seem reasonable.

So, to answer Soyer and Hogarth's question, we can look at the graph, and find the smallest value of x for which about 95% of the y-values are greater than 0. To me it looks like there are about 15 data points for each value of x, so to make things easier I'll call that 20 data points for each value of x. Then we just need to find the first point on the x-axis where, from that point on, there's no more than one y-value below 0 for any x. I'd say that's about $x = 50$. That's pretty close to 47, which is the approximate right answer according to Soyer and Hogarth. (Note that my counting was not so good: based on the method of data generation, there are probably only 10 data points for each value of x. But the method still worked well enough...)

3 Answering the question using just the summary statistics and the regression output

We should also be able to get the answer using just the summary statistics given above and the numbers from the regression output. So let's look at the regression output for our sample:

```
summM2


Call:
lm(formula = y ~ x, data = s2)

Residuals:
    Min      1Q  Median      3Q     Max
-100.93  -19.86   -0.83   19.42   86.11

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   0.2070     1.8071    0.11     0.91
x             1.0063     0.0316   31.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 28.7 on 998 degrees of freedom
Multiple R-squared: 0.504,Adjusted R-squared: 0.504
F-statistic: 1.02e+03 on 1 and 998 DF,  p-value: <2e-16


# get values to use in table
const   <- coef(summM2)[1, 1]
constSE <- coef(summM2)[1, 2]
xCoef   <- coef(summM2)[2, 1]
xCoefSE <- coef(summM2)[2, 2]
m2Rsq   <- summM2$r.squared
```

We can't expect all the numbers to match Soyer and Hogarth's exactly, but they're pretty close (see Table 3).

3.1 My approach

To answer Soyer & Hogarth's question using just the numbers in Table 1 and Table 3, we can use the formula for a $100(1-\alpha)\%$ prediction interval for the value of a new observation

Quantity	Soyer and Hogarth	LS
X	1.001 (0.033)	1.0063 (0.0316)
Constant	0.32 (1.92)	0.207 (1.8071)
R^2	0.50	0.5042
N	1000	1000
$\hat{\sigma}_e$ (= RMSE)	29	28.6968

Table 3: Comparison of regression outputs

Y_0 at $x = x_0$. Here's Casella & Berger's version of the formula:

$$\hat{\beta}_0 + \hat{\beta}_1 x_0 \pm t_{n-2,\alpha/2} S \sqrt{1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}}$$

That is, we can use the formula if certain conditions are met, or if we are willing to make the assumptions mentioned in the first section (this time with the definite inclusion of normality of the errors). Soyer and Hogarth do tell us to assume that the model "is indeed a very good approximation of the real world relation between X and Y, and that the linear estimation is suitable. Furthermore, among alternative specifications, this model is the one that gives the highest R-squared."

To find our x_0, we could set the lower bound of the 95% CI equal to 0, and plug in all the other quantities. We have $\hat{\beta}_0$, $\hat{\beta}_1$, n and S (that's the same as $\hat{\sigma}_e$) from Table 3, and we know $\bar{x}$ fromTable 1. Instead of using $t_{n-2,\alpha/2}$, we can use $Z_{\alpha/2}$, because the sample is so big. So that just leaves S_{xx}, which is equal to $\sum_{i=1}^{n}(x_i - \bar{x})^2$, or $(n-1)$ times the sample variance of x, which we can get from Table 1.

Solving for x_0 would take a lot of algebra, though. So let's try it the easy way instead, and just look for integer x_0's using R:

```
Sxx <- sd(s2$x)^2 * (n - 1)
foundX0 <- FALSE
x0 <- 0
while(!foundX0){
    SE <- m2Sigma * sqrt(1 + (1 / n) + ((x0 - mean(s2$x))^2 / Sxx))
    LB <- const + xCoef * x0 - qnorm(.975) * SE
    if(LB >= 0){
        print(x0)
        foundX0 <- TRUE
    } else{
        x0 <- x0 + 1
    }
}
```

```
[1] 56
```

But my answer, 56, is much higher than Soyer & Hogarth's answer of 47. Looking at their paper, I think I see why: they used a one-sided 95% prediction interval. Doing it this way

hadn't occurred to me, but it seems to make sense here because it corresponds to the visual approach I used with the scatterplot: I put all the probability in the lower tail. Here's what happens when I use a one-sided prediction interval:

```
foundX02 <- FALSE
x02 <- 0
while(!foundX02){
    SE <- m2Sigma * sqrt(1 + (1 / n) + ((x02 - mean(s2$x))^2 / Sxx))
    LB <- const + xCoef * x02 - qnorm(.95) * SE
    if(LB >= 0){
        print(x02)
        foundX02 <- TRUE
    } else{
        x02 <- x02 + 1
    }
}
```

```
[1] 47
```

So this time my answer, 47, matches their answer. But I'm left wondering: in reality does it make sense to use a one-sided interval here?

3.2 Soyer and Hogarth's approach

Soyer and Hogarth actually used a different approach to solve the problem. They noted that for any value of x, the residuals are distributed as $N(0, SER^2)$, where SER is another name for $RMSE$ or $\hat{\sigma}_e$ and is equal to 29 (or in our case, 28.6968). They looked for the specific value of x for which only 5% of the residuals would be expected to be longer in the same direction than the difference between zero and the fitted value for that x. That is, they looked for the x that would satisfy

$$P(\hat{e} < 0 - (\hat{\beta}_0 + \hat{\beta}_1 x)) = .05$$

or, if we standardize,

$$P\left(Z < \frac{[0 - (\hat{\beta}_0 + \hat{\beta}_1 x)] - 0}{SER}\right) = .05$$

but we know that $P(Z < -1.645) = .05$, so we have

$$\begin{aligned}
\frac{-(\hat{\beta}_0 + \hat{\beta}_1 x)}{SER} &= -1.645 \\
-(\hat{\beta}_0 + \hat{\beta}_1 x) &= -1.645 * SER \\
-\hat{\beta}_1 x &= -1.645 * SER + \hat{\beta}_0 \\
x &= \frac{-1.645 * SER + \hat{\beta}_0}{-\hat{\beta}_1} \\
x &= \frac{1.645 * SER - \hat{\beta}_0}{\hat{\beta}_1}
\end{aligned}$$

Solving in R...

```
x1 <- ((qnorm(.95) * m2Sigma) - const) / xCoef
x2 <- ((qnorm(.975) * m2Sigma) - const) / xCoef
```

...we get $x = 46.6986$ if we use a one-sided approach as Soyer and Hogarth did, and $x = 55.6842$ if we use a two-sided approach. These values are consistent with the integer values I found above (47 and 56).

4 Answering the question using both the scatterplot and the numbers

Now let's go back to our original scatterplot with the regression line, and add in a symmetric 95% prediction interval. We can show the 95% confidence interval on the same plot just to show how much narrower it is. Because we're interested in the value of a single observation, we want to use the prediction interval, not the confidence interval. I'll show two different ways of doing this in R:

```
# prepare for the ggplot
s2WithPred = data.frame(s2, predict(m2, interval = 'prediction'))
```

```
Warning:  Predictions on current data refer to _future_ responses
```

```
# prepare for the base R plot
newxs <- seq(0, 99, by=1)
c1 <- data.frame(predict(m2, newdata= data.frame(x= newxs),
    interval = 'confidence'))
c2 <- data.frame(predict(m2, newdata= data.frame(x= newxs),
    interval = 'prediction', level= .90))
c3 <- data.frame(predict(m2, newdata= data.frame(x= newxs),
    interval = 'prediction', level= .95))
```

```
# make ggplot w/ regression line, confidence interval and prediction interval
ggplot(s2WithPred, aes(x= x, y= y)) +
    geom_point() +
    geom_smooth(method= 'lm', aes(fill= 'confidence'),
        alpha= 0.5) +
    geom_ribbon(aes(y= fit, ymin= lwr, ymax= upr,
        fill= 'prediction'), alpha= 0.2) +
    scale_fill_manual('Interval', values= c('green', 'blue')) +
    opts(legend.position= c(0.20, 0.85)) +
    guides(fill= guide_legend(
        override.aes= list(alpha = c(0.5, 0.2))))
```

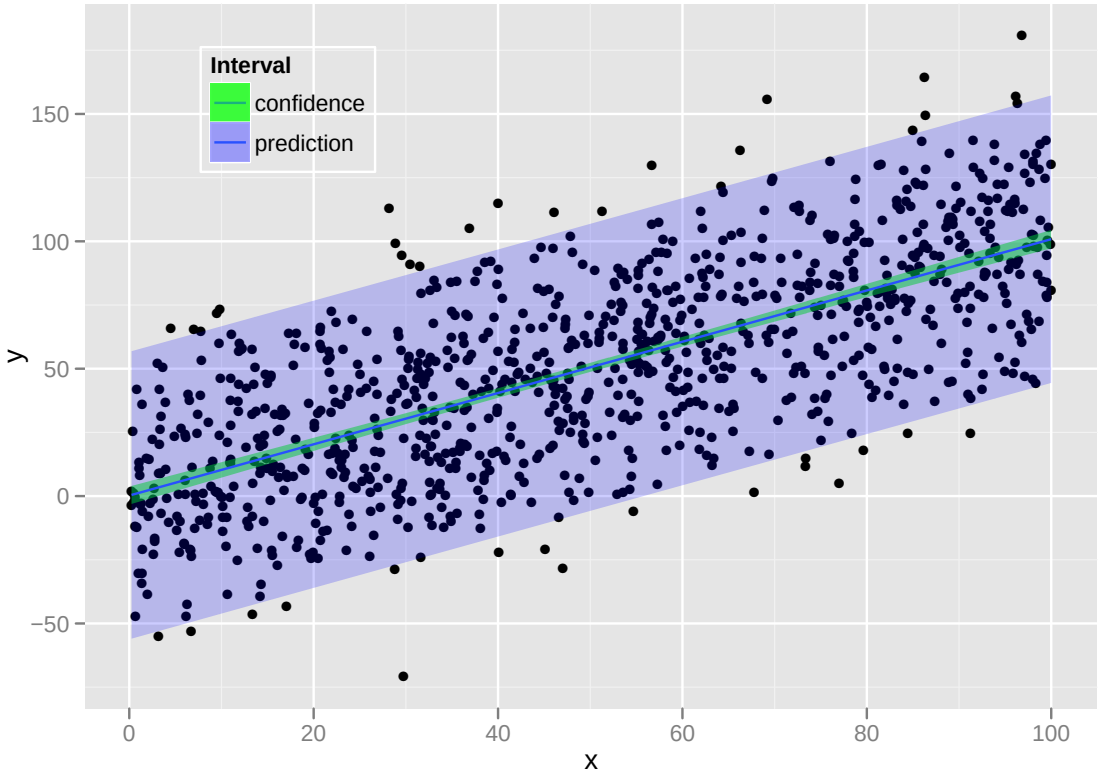

Figure 3: One way to plot this in R (ggplot2 graphics).

```r
# make base R plot w/ regression line, confidence interval,
# and two prediction intervals
with(s2, plot(x, y, xlim= c(0,100), xaxp= c(0, 100, 20), ylim= c(-100, 200)))
abline(h= 0, col= "grey50", lty= 1)
abline(m2, lwd= 2)
lines(cbind(newxs, c1$lwr), col= "red", lwd= 2, lty= 2)
lines(cbind(newxs, c1$upr), col= "red", lwd= 2, lty= 2)
lines(cbind(newxs, c2$lwr), col= "purple", lwd= 2, lty= 4)
lines(cbind(newxs, c3$lwr), col= "blue", lwd= 2, lty= 3)
lines(cbind(newxs, c3$upr), col= "blue", lwd= 2, lty= 3)
legend("topleft",
    legend= c("95% CI", "Symmetric 95% Pred. Int.", "LB of 1-sided 95% Pred. Int."),
    lty = c(2, 3, 4),
    col= c("red", "blue", "purple")
)
```

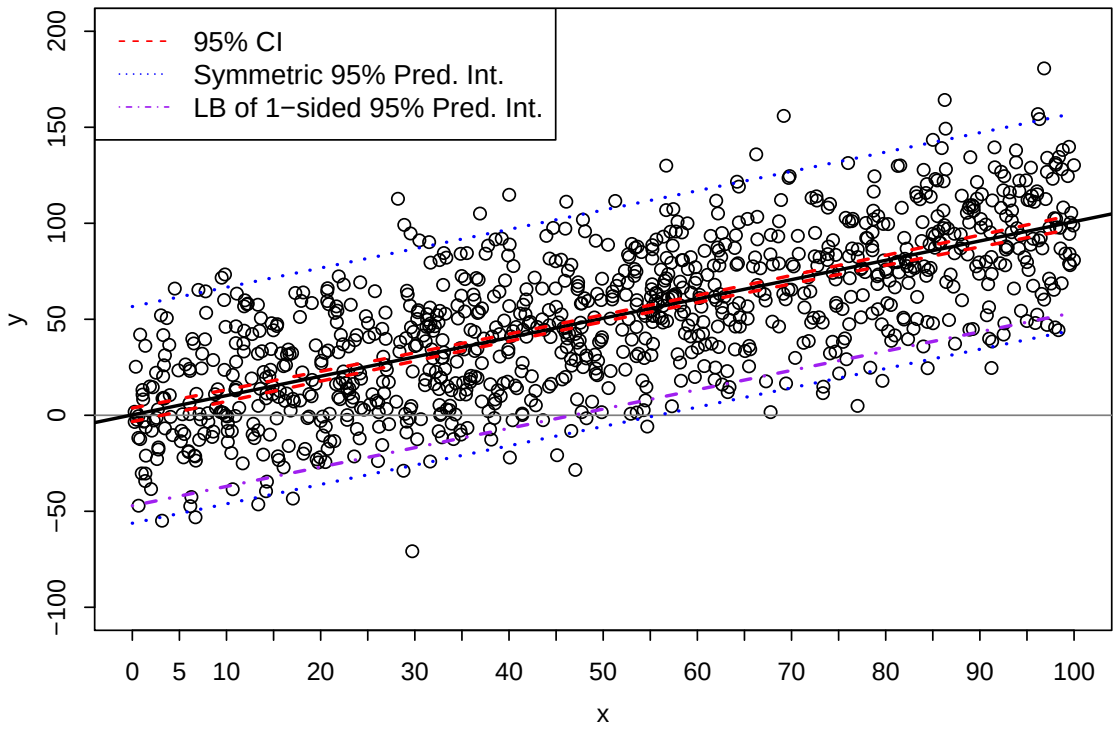

Figure 4: Another way to plot this in R (base R graphics).

Both plots provide visual confirmation our answer using the symmetric two-sided approach; in addition, Figure 4 confirms our answer using a one-sided approach.

5 The nonreproducible part: how to do this in Stata

What if we wanted to do all of this in Stata instead? Using Stata 12, here is code with output:

```
clear
set seed 20120712
drawnorm Errs, n(1000) means(0) sds(29)
gen Xs = 100 * runiform()
*
**************************************
summarize Xs

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          Xs |      1000    48.66937    29.52073   .0633842   99.96919


gen Ys = 1 * Xs + Errs
*
**************************************
summarize Ys
   Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          Ys |      1000    49.98331    41.80034  -56.43404   162.1941

**************************************
*
regress Ys Xs
      Source |       SS       df       MS              Number of obs =    1000
-------------+------------------------------           F(  1,   998) = 1139.74
       Model |   930626.71     1   930626.71           Prob > F      =  0.0000
    Residual |  814894.263   998  816.527317           R-squared     =  0.5332
-------------+------------------------------           Adj R-squared =  0.5327
       Total |  1745520.97   999  1747.26824           Root MSE      =  28.575

------------------------------------------------------------------------------
          Ys |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          Xs |   1.033898    .030625    33.76   0.000     .9738017    1.093995
       _cons |  -.3358742   1.743017    -0.19   0.847    -3.756273    3.084525
------------------------------------------------------------------------------
*
*
*
```

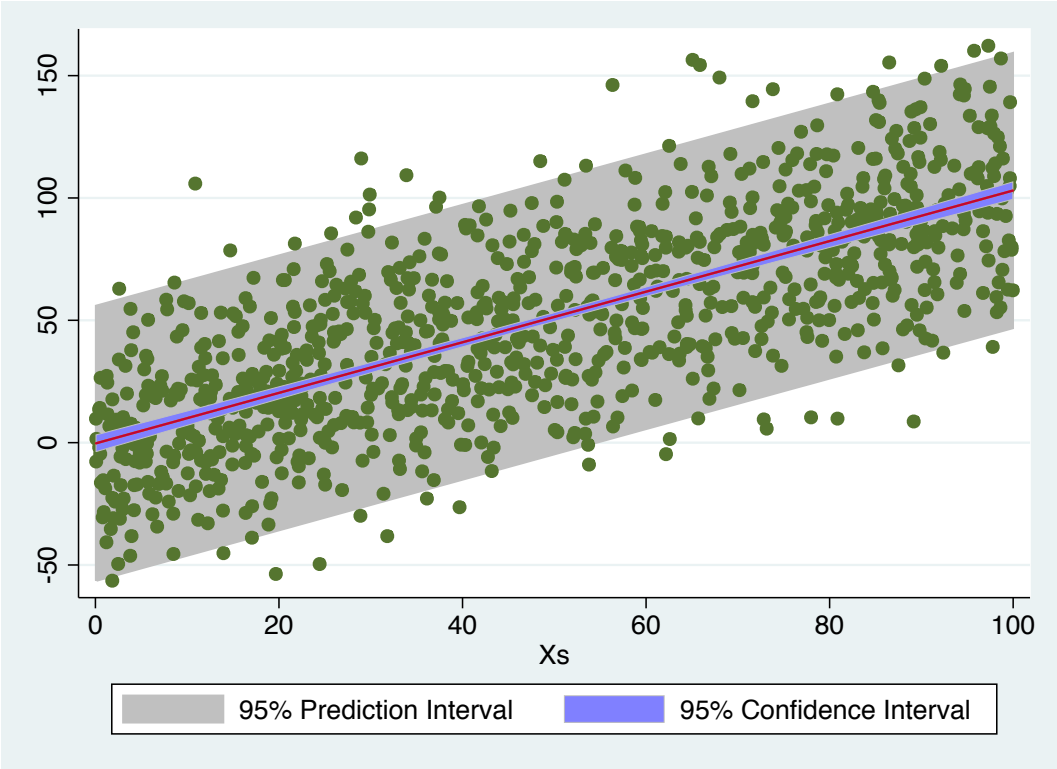

Figure 5: The graph from Stata.

```
* line breaks are for presentation only-- paste into Stata on one line!
graph twoway
    (lfitci Ys Xs, level(95) stdf ciplot())
    (scatter Ys Xs)
    (lfitci Ys Xs,
        level(95)  fcolor(blue) alwidth(none) fintensity(50) ciplot())
    (lfit Ys Xs),
    legend(order(1 4) label(1 "95% Prediction Interval")
        label(4 "95% Confidence Interval"))
```

6 Session Information: R/Package Versions

```
                _
version.string R version 2.15.1 (2012-06-22)

          Version                                Depends
colorout    0.9-9                                  <NA>
ggplot2     0.9.1          R (>= 2.14), stats, methods
gridExtra     0.9                    R(>= 2.4.0), grid
Hmisc       3.9-3      R (>= 2.4.0), methods, survival
knitr         0.7                         R (>= 2.14.1)
plyr        1.7.1                         R (>= 2.11.0)
rms         3.5-0 Hmisc (>= 3.7), survival (>= 2.36-3)
setwidth    1.0-0                                  <NA>
vimcom      0.9-2                                  <NA>
```