

MATH 204: PRINCIPLES OF STATISTICS 2

WINTER 2026

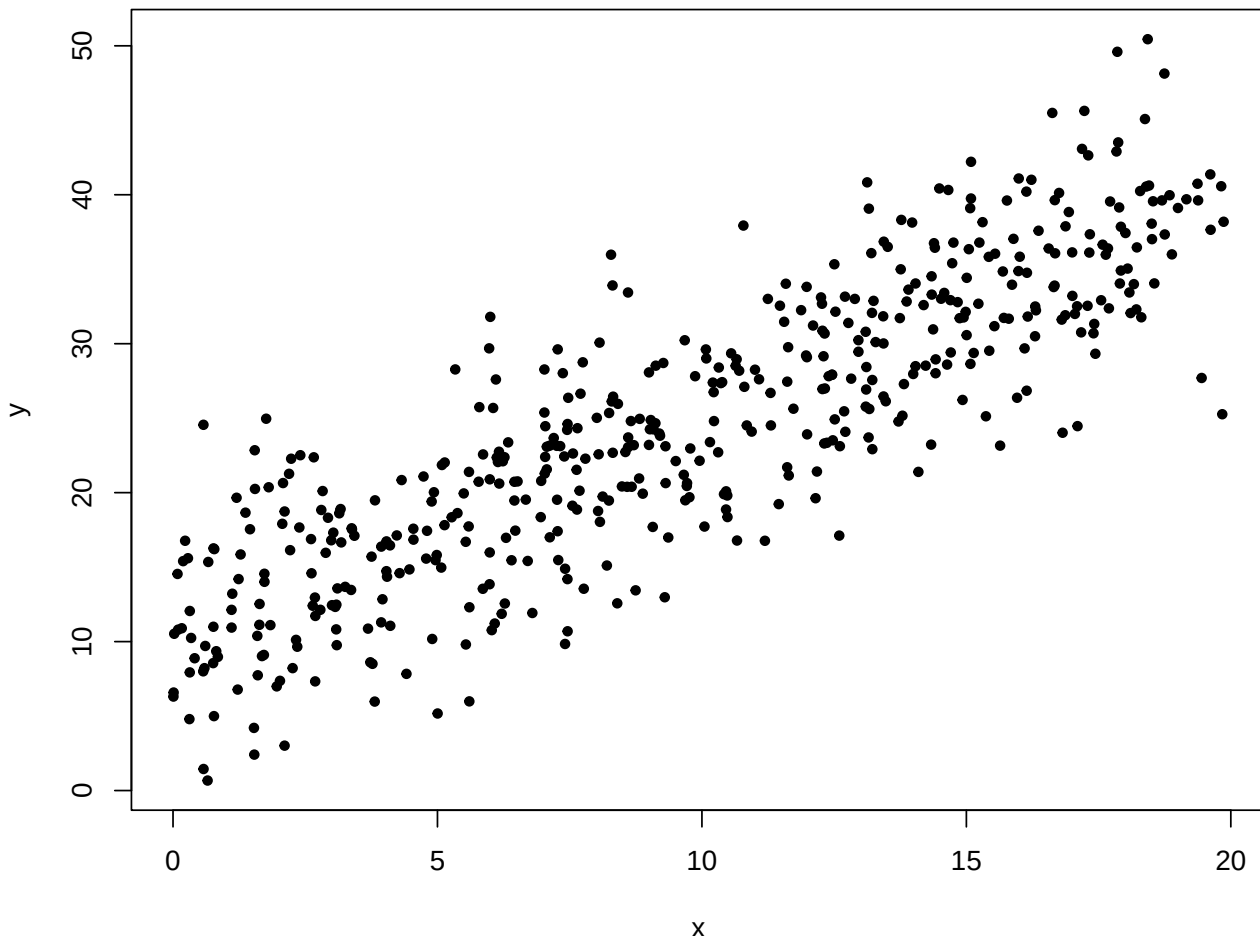
SIMPLE LINEAR REGRESSION: ESTIMATION AND PREDICTION

To illustrate some of the key calculations associated with simple linear regression, we consider a simulation study where data are generated from the following model: for $i = 1, \dots, n$

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where $\beta_0 = 10$, $\beta_1 = 1.5$ and $\sigma^2 = 5^2$. The random residual terms, ϵ_i , represent the measurement errors, and are assumed to be Normally distributed. Finally, the x_i s are selected uniformly from the range 0 to 20. Here, $n = 500$.

```
> set.seed(4590)
> n<-500
> x<-runif(n, 0, 20)
> b0<-10
> b1<-1.5
> sig<-5
> y<-b0+b1*x+rnorm(n)*sig
> par(mar=c(4, 4, 2, 2))
> plot(x, y, pch=19, cex=0.7)
```



We can obtain the parameter estimates using the `lm` function.

```

> summary(lm(y~x))

+
+ Call:
+ lm(formula = y ~ x)
+
+ Residuals:
+      Min       1Q   Median       3Q      Max
+ -14.358  -3.280   0.227   3.056  13.622
+
+ Coefficients:
+              Estimate Std. Error t value Pr(>|t|)
+ (Intercept)  10.0760     0.4452   22.6    <2e-16 ***
+ x             1.4894     0.0399   37.3    <2e-16 ***
+ ---
+ Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
+
+ Residual standard error: 4.97 on 498 degrees of freedom
+ Multiple R-squared:  0.736, Adjusted R-squared:  0.736
+ F-statistic: 1.39e+03 on 1 and 498 DF,  p-value: <2e-16

```

That is, the estimates $(\hat{\beta}_0, \hat{\beta}_1)$ are (10.0760, 1.4894). In the standard error column, which is headed Std. Error, we see measures of variability of the corresponding estimators. Recall the following terminology:

- the *estimate* is a **numerical value**, obtained as a function or summary of the observed data, that gives us the line of best fit;
- the *estimator* is a **random variable** which is obtained by applying the same function used for the estimate to the response random variables;
- a random variable has a **probability distribution**, and for an estimator, this probability distribution is called the *sampling distribution*;
- the sampling distribution describes how the estimator would behave in **repeated runs** of the experiment under identical conditions;
- the *standard error* is simply the **standard deviation** of the sampling distribution;
- in this case, as we will use an estimate of σ , we actually compute the *estimated standard error* (e.s.e.) for each estimator.

The joint sampling distribution of the two estimators in this case is quite complicated. We typically focus on the sampling distribution of each estimator *separately*, and in this case, the two sampling distributions are both t-distributions with $n - 2$ degrees of freedom, centered at the true value of the parameter in each case. This allows us to write down the confidence interval at level $100(1 - \alpha)\%$ as

$$\hat{\beta}_j \pm t_{\alpha/2, n-2} \times \text{e.s.e.}(\hat{\beta}_j) \quad j = 0, 1.$$

This confidence interval is simply a high-probability region of the sampling distribution.

In the coefficients table, we also have the t value column, where the values are computed by the formula

$$t \text{ value} = \frac{\text{Estimate}}{\text{Std. Error}}$$

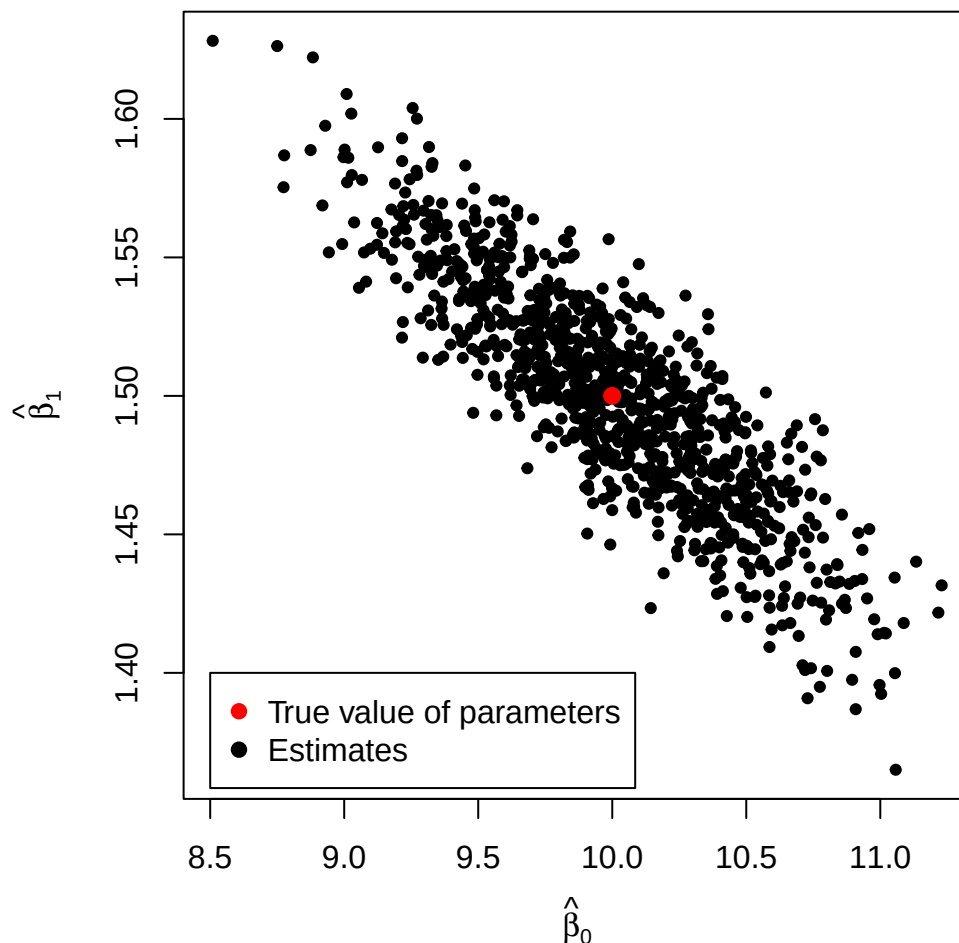
and the p-value in the test of the null hypothesis that the true value of the parameter is zero.

This discussion emphasizes that, even though we have a large sample, we are still not absolutely certain what the true underlying straight line is; we only have a line of best fit that is our best guess at the true line based on the available data. The sampling distribution is a representation of how variable the estimates would be on repeated implementations of the experiment for a sample of the same size. We can examine the sampling distribution using a simulation study: we replicate the data generation mechanism re-generating the response variable each time

```
> nreps<-1000
> beta.mat<-matrix(0,nrow=nreps, ncol=2)
> for(irep in 1:nreps){
+   yr<-b0+b1*x+rnorm(n)*sig
+   beta.mat[irep,]<-coef(lm(yr~x))
+ }
```

We can the plot the resulting set of $n = 1000$ estimates on a scatterplot.

```
> par(mar=c(6,6,1,1))
> plot(beta.mat,pch=19, cex=0.7,
+       xlab=expression(hat(beta)[0]), ylab=expression(hat(beta)[1]))
> points(b0,b1,pch=19,cex=1.1,col='red')
> legend(8.5,1.4,c('True value of parameters','Estimates'),
+        pch=19,col=c('red','black'))
```



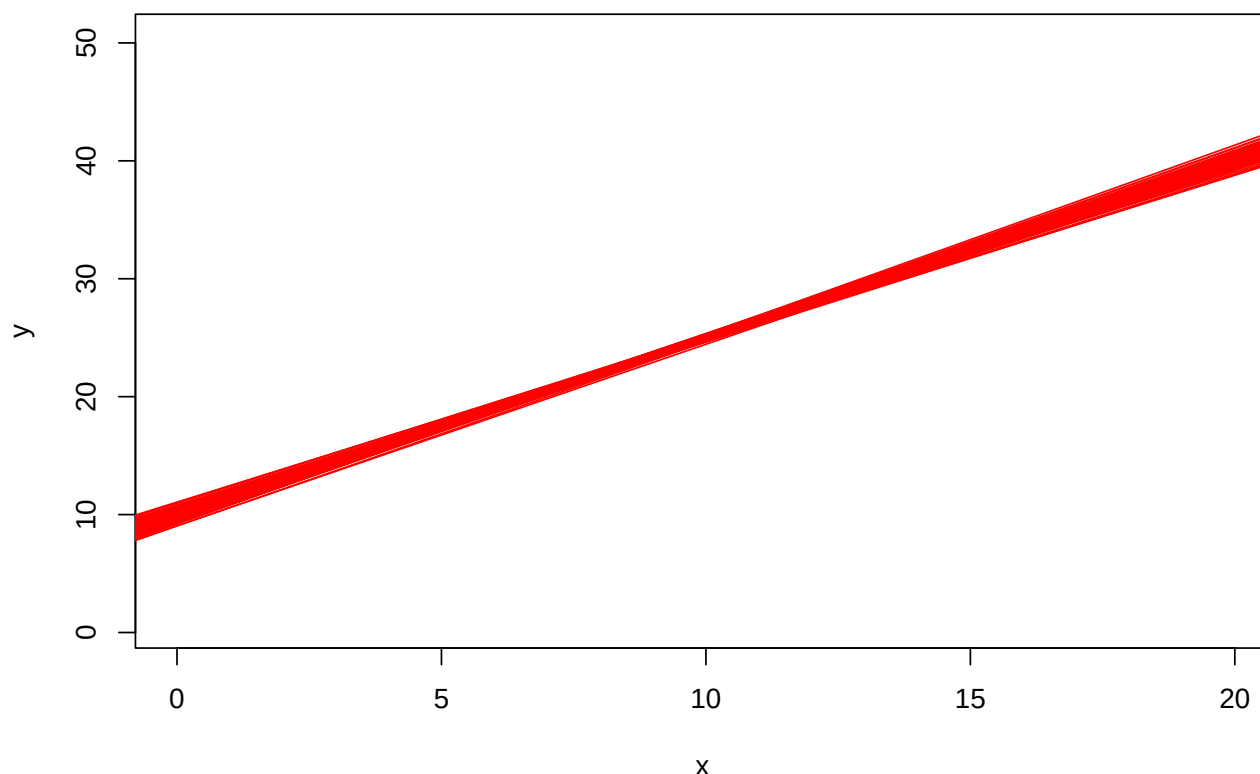
This is a representation of the sampling distribution, and shows the variability of a sample of estimates obtained for a sample size of $n = 500$. Clearly the distribution of the estimator is centered at the true value. If we compute the standard deviation of the sample of estimates, they match quite closely the entries in the s.e. column of the coefficients table.

```
> c(sd(beta.mat[,1]),sd(beta.mat[,2])) #Results of the simulation study
+ [1] 0.4504606 0.0403616

> summary(lm(y~x))$coef[,2]           #Standard errors from fit to data
+ (Intercept)          x
+ 0.4452270 0.0399318
```

Another way to represent the variability is to plot out a sample of the lines of best fit from the simulation study:

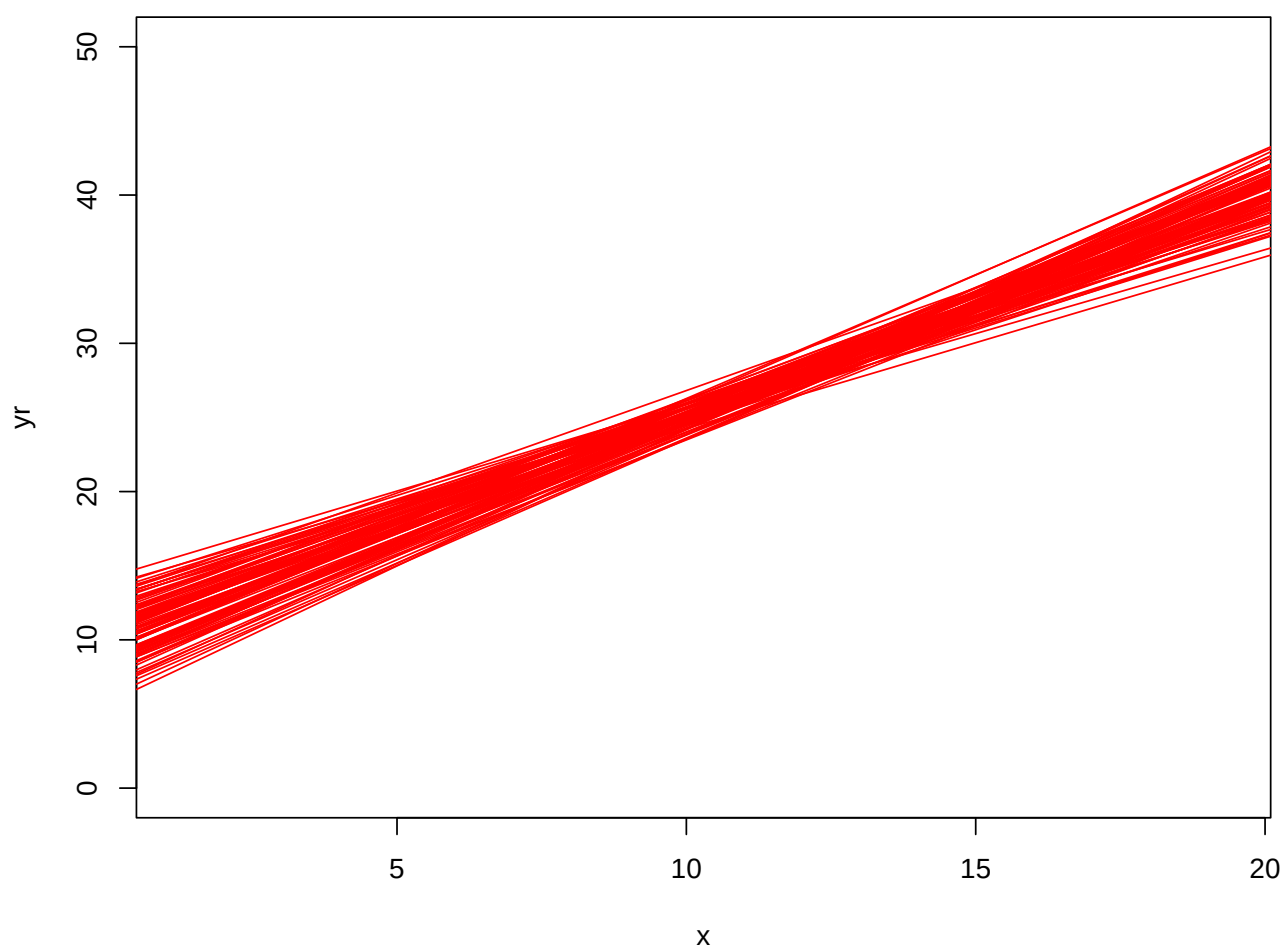
```
> par(mar=c(4,4,2,2))
> plot(x,y,type='n')
> for(i in 1:100){abline(beta.mat[i,],col='red')}
```



The red band shows the high probability region for the line of best fit that would arise from repeated calculations for replicate data sets.

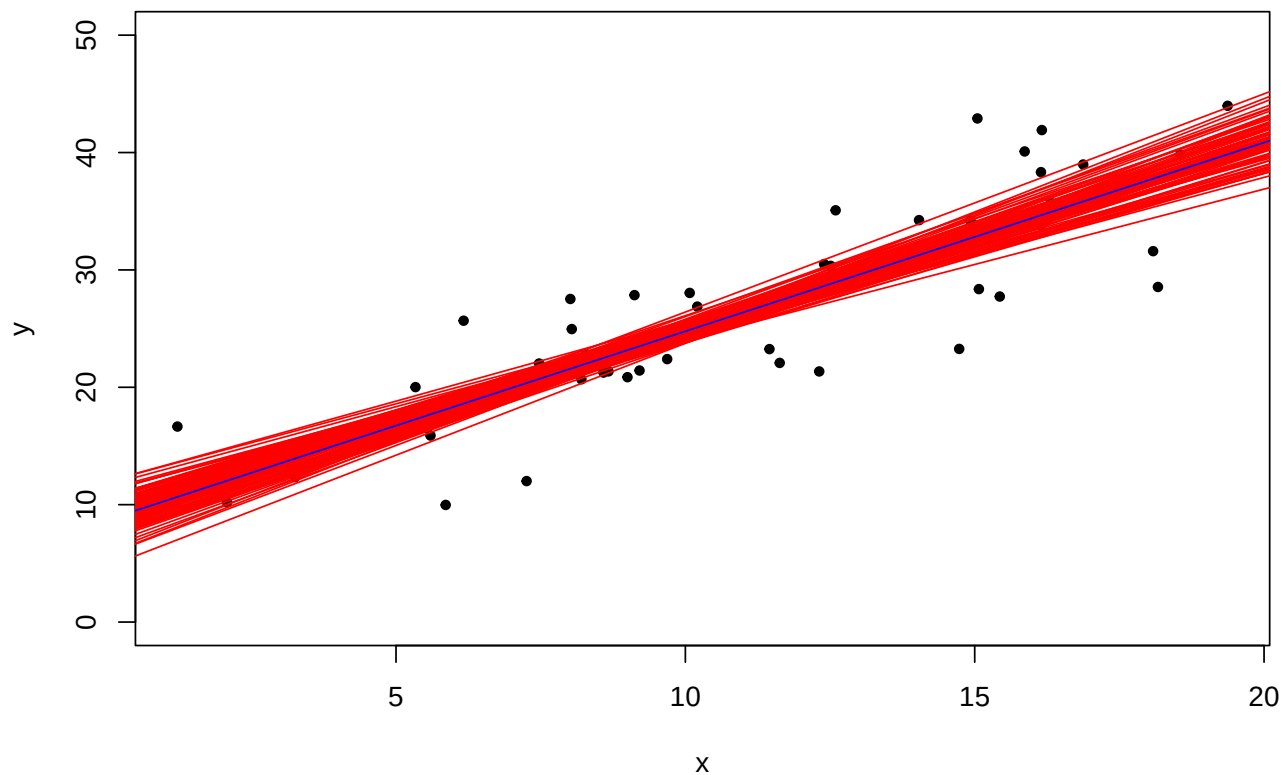
With a smaller data set of $n = 50$, the uncertainty would be greater.

```
> set.seed(4590)
> n<-50
> x<-runif(n,0,20)
> beta.mat<-matrix(0,nrow=nreps, ncol=2)
> for(irep in 1:nreps){
+   yr<-b0+b1*x+rnorm(n)*sig
+   beta.mat[irep,]<-coef(lm(yr~x))
+ }
> par(mar=c(4,4,2,2))
> plot(x,yr,type='n',ylim=range(0,50))
> for(i in 1:100){abline(beta.mat[i,],col='red')}
```



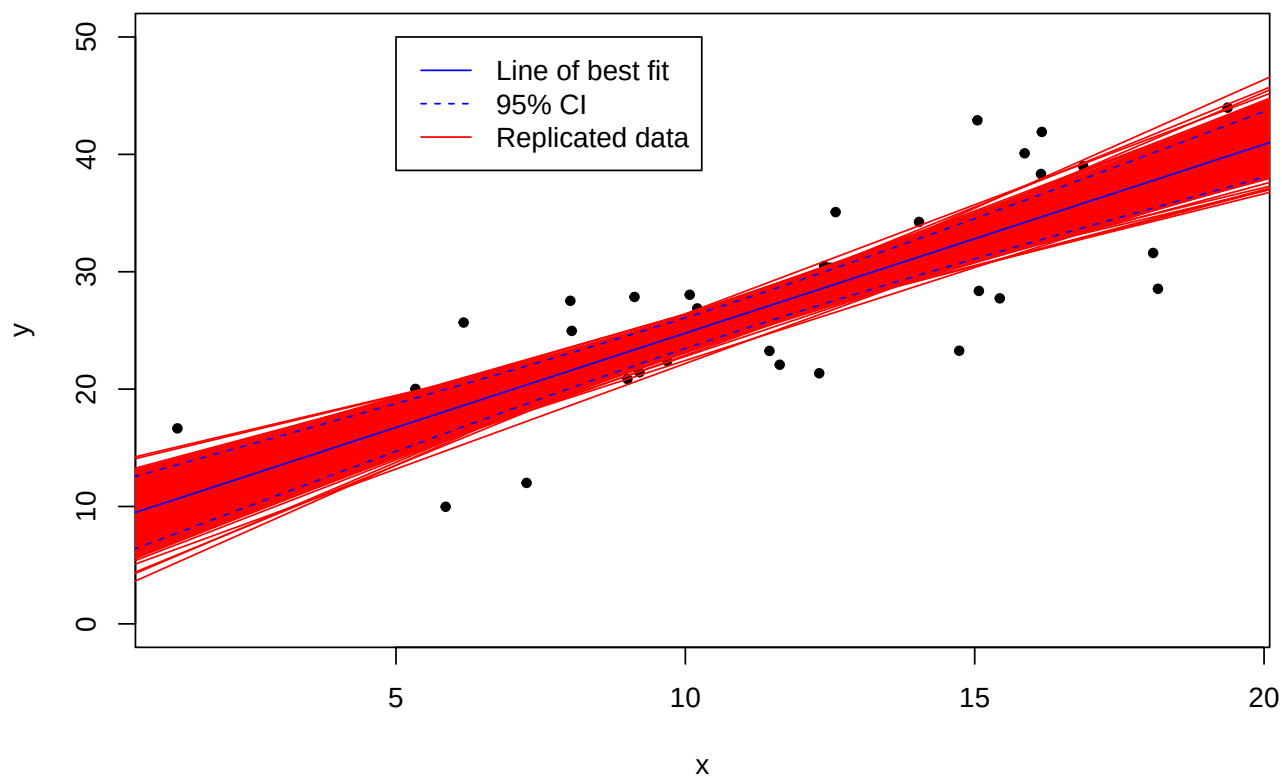
To represent this variability for a *given* data set, this simulation approach needs to be modified. Rather than sampling new residual errors in the replicate data sets, we instead re-sample the *residuals* from the fit of the model, and the parameter estimates. This approach is known as *bootstrapping*.

```
> set.seed(4590)
> n<-50
> x<-runif(n,0,20)
> y<-b0+b1*x+rnorm(n)*sig
> par(mar=c(4,4,2,2))
> plot(x,y,pch=19,cex=0.7,ylim=range(0,50))
> f1<-lm(y~x)
> b0.hat<-coef(f1)[1]
> b1.hat<-coef(f1)[2]
> resids<-residuals(f1)
> beta.mat<-matrix(0,nrow=nreps, ncol=2)
> for(irep in 1:nreps){
+   yr<-b0.hat+b1.hat*x+sample(resids,size=n,rep=T)
+   beta.mat[irep,]<-coef(lm(yr~x))
+ }
> for(i in 1:100){abline(beta.mat[i,],col='red')}
> abline(coef(f1),col='blue')
```



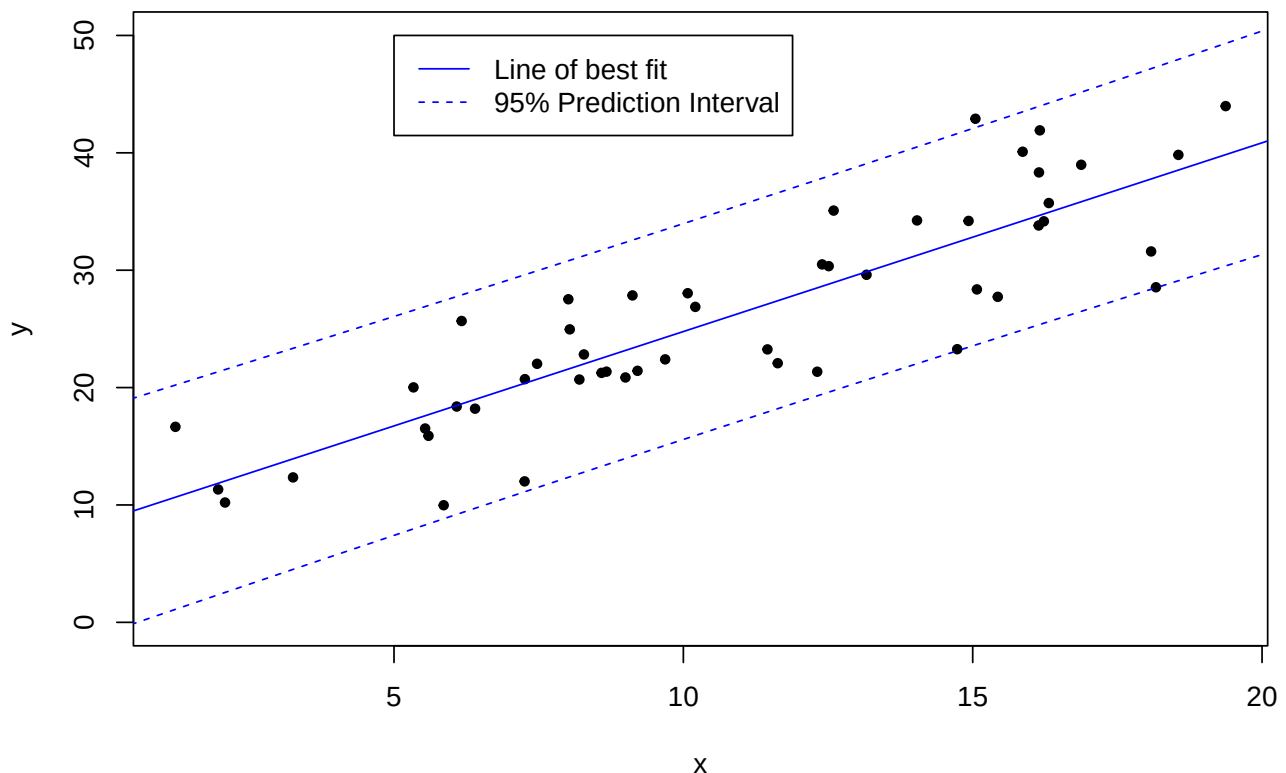
These samples of lines of best fit from bootstrap replicated data sets capture a similar sort of variability to the prediction intervals computed by the `predict` function.

```
> set.seed(4590)
> n<-50
> x<-runif(n,0,20)
> y<-b0+b1*x+rnorm(n)*sig
> par(mar=c(4,4,2,2))
> plot(x,y,pch=19,cex=0.7,ylim=range(0,50))
> for(i in 1:nreps){abline(beta.mat[i,],col='red')}
> f1<-lm(y~x)
> abline(coef(f1),col='blue')
> xv<-seq(0,20,by=0.01)
> yhat<-predict(f1,newdata=data.frame(x=xv),interval='confidence')
> lines(xv,yhat[,2],col='blue',lty=2)
> lines(xv,yhat[,3],col='blue',lty=2)
> legend(5,50,c('Line of best fit','95% CI','Replicated data'),
+         col=c('blue','blue','red'),lty=c(1,2,1))
```



Finally, this form of prediction interval only represents the variability in the line of best fit, and does not take into account the measurement error that is present when we make a new measurement of the response. AN interval that does take this into account can also be computed using the `predict` function, using the `interval='predict'` argument.

```
> par(mar=c(4, 4, 2, 2))
> plot(x, y, pch=19, cex=0.7, ylim=range(0, 50))
> f1<-lm(y~x)
> abline(coef(f1), col='blue')
> xv<-seq(0, 20, by=0.01)
> yhat<-predict(f1, newdata=data.frame(x=xv), interval='prediction')
> lines(xv, yhat[, 2], col='blue', lty=2)
> lines(xv, yhat[, 3], col='blue', lty=2)
> legend(5, 50, c('Line of best fit', '95% Prediction Interval'),
+       col=c('blue', 'blue'), lty=c(1, 2, 1))
```



Notice that for both types of interval, the prediction interval gets **wider** the further the prediction point is from the mean of the x values.