Bootstrap Variations

Contents


```
sharks <- read.csv("../Data/Sharks/sharks.csv")
popSharks <- rownames(sharks)

#samples <- combn(popSharks, 5)
#N_s <-  ncol(samples)
N_s <- 10^4
n = 6
set.seed(341)
samples <- sapply(1:N_s, FUN =function(b)  sample(popSharks, n, replace = TRUE) )

avePop <- mean(sharks[, "Length"])
avesSamp <- apply(samples, MARGIN = 2,
                  FUN = function(s){mean(sharks[s,"Length"])})
sampleErrors <- avesSamp - avePop

tmpAve <- mean(avesSamp)
tmpSD <- sd(avesSamp)
```

```r
sdsSamp <- apply(samples, MARGIN = 2,
                 FUN = function(s){sd(sharks[s,"Length"])})
```

```r
directory <- "../Data"
dirsep <-"/"
filename <- paste(directory, "agpop_data.csv", sep=dirsep)
agpop <- read.csv(filename, header=TRUE)

missing92 <- agpop[,"acres92"] == -99
rowNumsMissing <- which(agpop[,"acres92"] == -99)
agpop[missing92, "acres92"] <- NA
agpop[agpop[,"acres87"] == -99, "acres87"] <- NA
agpop[agpop[,"acres82"] == -99, "acres82"] <- NA
```

Bootstrap

- So far, to bootstrap we have been sampling with replacement from the sample $\mathcal{S}$.
  - Because the sample $\mathcal{S}$ was viewed as an estimate of the population $\mathcal{P}$
  - This sampling scheme is equivalent to sampling from the empirical distribution function, $\widehat{F}$.

- In other words, we would like to sample from the distribution F, but instead,
  - we obtain a sample using an estimate $\widehat{F}$.

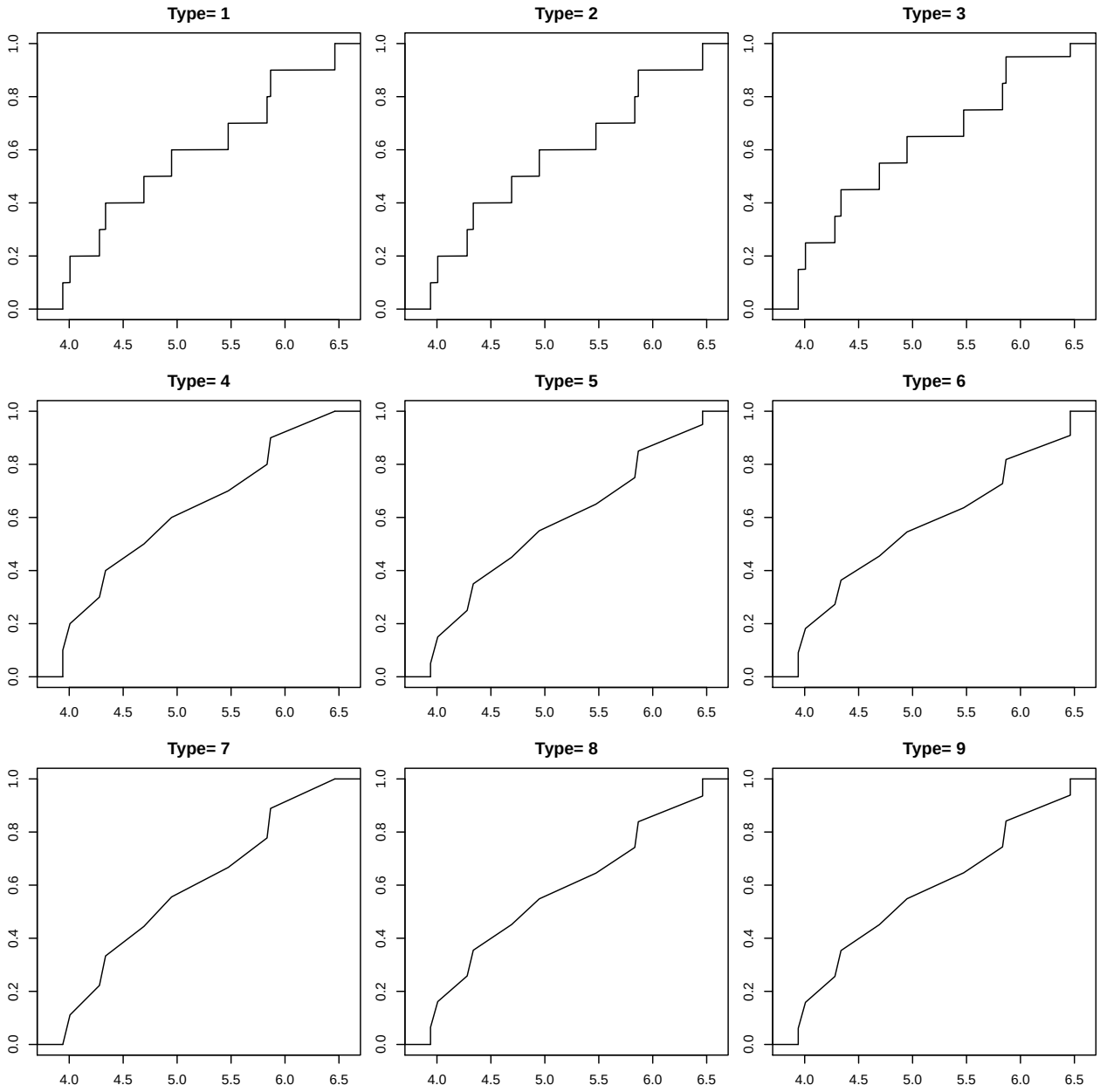
- What other possible estimates are there for the cumulative distribution function F?

Estimate for F

- Varies empirical distribution functions using the argument `type` in the `quantile` function.
  - Generate 10 observations from $G(5, 1)$

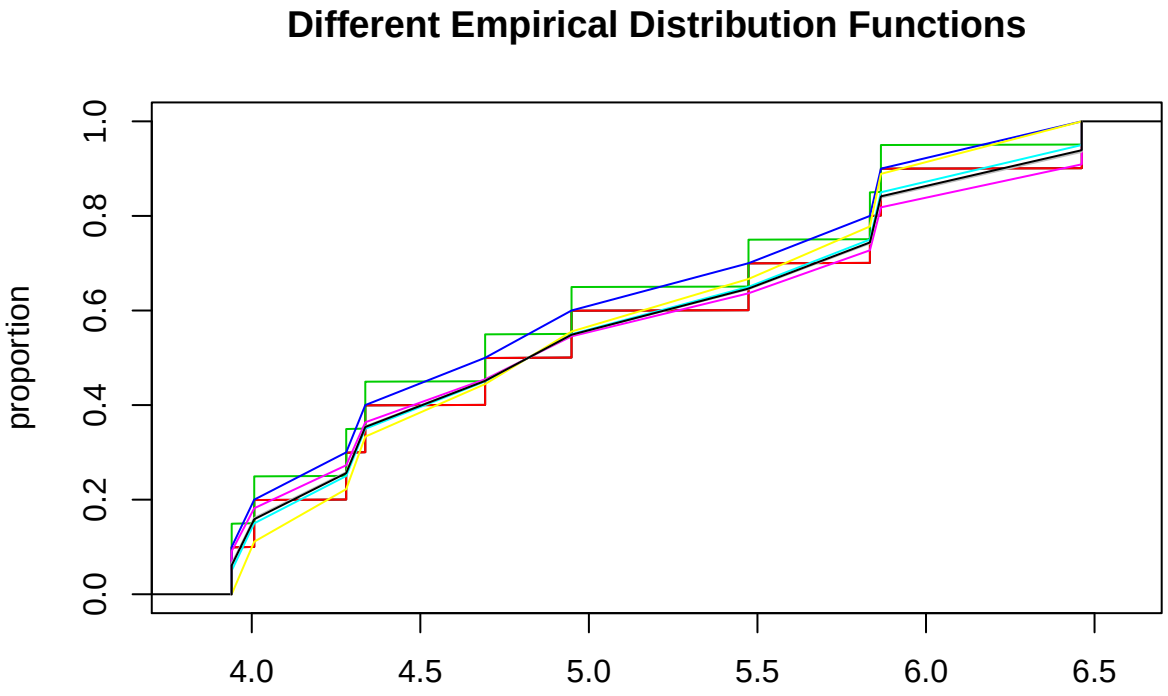
```r
set.seed(341)
x = rnorm(10, mean=5)
```

```r
pseq = seq(0, 1, length.out=1000)
par(mfrow=c(3,3), mar=2.5*c(1,1,1,0.1))
for (i in 1:9) {
  plot( quantile(x, probs=pseq, type=i), pseq, type='l', xlim= extendrange(x),
        ylim=c(0,1), ylab="proportion", xlab="", main=paste("Type=", i) )
  segments(x0 = c(-5,10), y0 = c(0,1), x1 = c(min(x), max(x)), y1 = c(0,1))
  }
```



- Refer to the help documentation of the `quantile` function for details on the argument `type`.

- All on the quantile functions on one plot



Parameteric Estimate

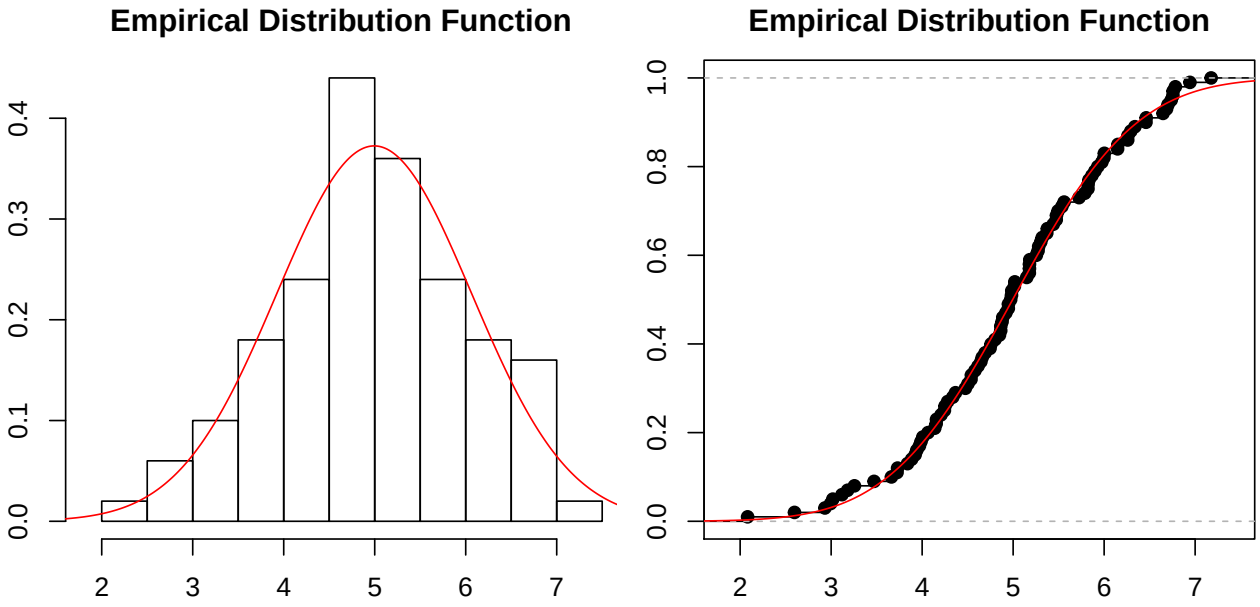
- We can estimate the distribution function $F(x)$ using a parametric model $F(x;\theta)$ which is indexed by some parameters.
- Generate 100 observations from $G(\mu = 5, \sigma = 1)$

```
set.seed(341)
x = rnorm(100, mean=5)
c(mean(x), sd(x))
```

```
## [1] 4.994535 1.070854
```

- Overlay the fitted Gaussian distribution $G(\mu = \widehat{\mu}, \sigma = \widehat{\sigma})$ on the empirical CDF.

```
par(mfrow=c(1,2), mar=2.5*c(1,1,1,0.1))
xseq= seq(-10, 10, length.out=1000)
hist( x, breaks="FD", xlim= extendrange(x),  ylab="proportion", xlab="",
      main="Empirical Distribution Function", prob=TRUE)
lines(xseq, dnorm(xseq, mean(x), sd=sd(x) ), col=2 )

plot( ecdf(x), xlim= extendrange(x), ylim=c(0,1), ylab="proportion", xlab="",
      main="Empirical Distribution Function")

lines(xseq, pnorm(xseq, mean(x), sd=sd(x) ), col=2 )
```



Non-parameteric Bootstrap

- For a given sample $\mathcal{S}$ and non-parametric method
  - Obtain an estimate $\widehat{F}(x)$ using the sample: this estimate is the empirical CDF

- Generate B bootstrap samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ using $\widehat{F}(x)$
  - when you sample with replacement from $\mathcal{S}$, you are generating samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ using $\widehat{F}(x)$

- **Note:** Alternatively, we could estimate the density function with some $\widehat{f}$, and do the same thing.

Parameteric Bootstrap

- For a given sample $\mathcal{S}$ and parametric model $F(x;\theta)$
  - Obtain an estimate $\widehat{\theta}$ using the sample

- Generate B bootstrap samples $\mathcal{S}_1^\star, \ldots, \mathcal{S}_B^\star$ using $F(x;\widehat{\theta})$.
  - Here, we will generate samples from the model, NOT through sampling with replacement from the sample.

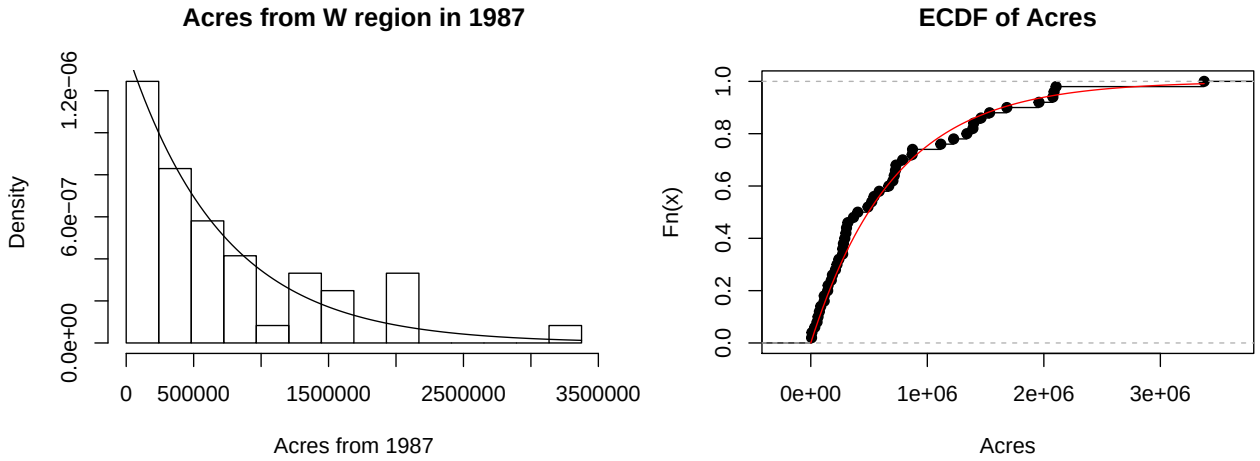
Agricultural census (USA): Parametric Example

- Consider the West region and suppose we obtain a sample of size 50 from the 422 farms in Western region and measure the number of acres in 1987

```
set.seed(341)
acres87 = agpop[agpop$region == "W","acres87"]
N = length(acres87)
n =50
acres87Sam = acres87[sample(1:N, n)]
```

- From a the histogram and empirical distribution function, it seems an exponential distribution with rate equal to $1/\overline{x}$ fits the data well.
  - note that $1/\overline{x}$ is the maximum likelihood estimate of the parameter of the exponential distribution.

```
par(mfrow=c(1,2))
hist(acres87Sam, breaks=seq(0, max(acres87Sam, na.rm=TRUE), length.out = 15) ,
     prob=TRUE, xlab="Acres from 1987", main="Acres from W region in 1987")

zseq = seq(0, max(acres87Sam, na.rm=TRUE), length.out = 100 )
lines(zseq, dexp(zseq, rate=1/mean(acres87Sam)))

plot( ecdf(acres87Sam), main="ECDF of Acres", xlab="Acres" )
lines(zseq, pexp(zseq, rate=1/mean(acres87Sam)), col=2, xlab="Acres" )
```



- The smooth curve is the fitted exponential model on the histogram/empirical CDF based on the data.
- Based on these graphs, exponential distribution seems to be a good fit for the data.
- This means that to repeat the experiment to get more samples each of size n, one can simply generate n random samples from $exp(1/\overline{x})$ distribution.

```
theta = 1/mean(acres87Sam)
B = 10^4
```

```
Sstar <- sapply(1:B, FUN =function(b) rexp(n, rate=theta))
bootAvg = apply(Sstar,2,mean)
```

The summary statistics for the sample averages of the **parametric** bootstrap sample are

```
sd(bootAvg) ; summary(bootAvg)
```

```
## [1] 99363.83
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  385665  644119  708854  713696  777726 1157148
```

Agricultural census (USA): Non-parametric Example

- Consider the West region and suppose we obtain a sample of size 50 from the 422 farms in Western region and measure the number of acres in 1987
- From the sample of size 50 we take B samples with replacement, each of size n,
  - we calculate the average on each of the B sampels
  - then we look at the summary statistics of these averages

The summary statistics for the sample averages of the **non-parametric** bootstrap sample are

```
Sstar0 <- sapply(1:B, FUN =function(b) acres87Sam[sample(n,n, replace=TRUE)] )
bootAvg0 = apply(Sstar0,2,mean)
sd(bootAvg0) ; summary(bootAvg0)
```

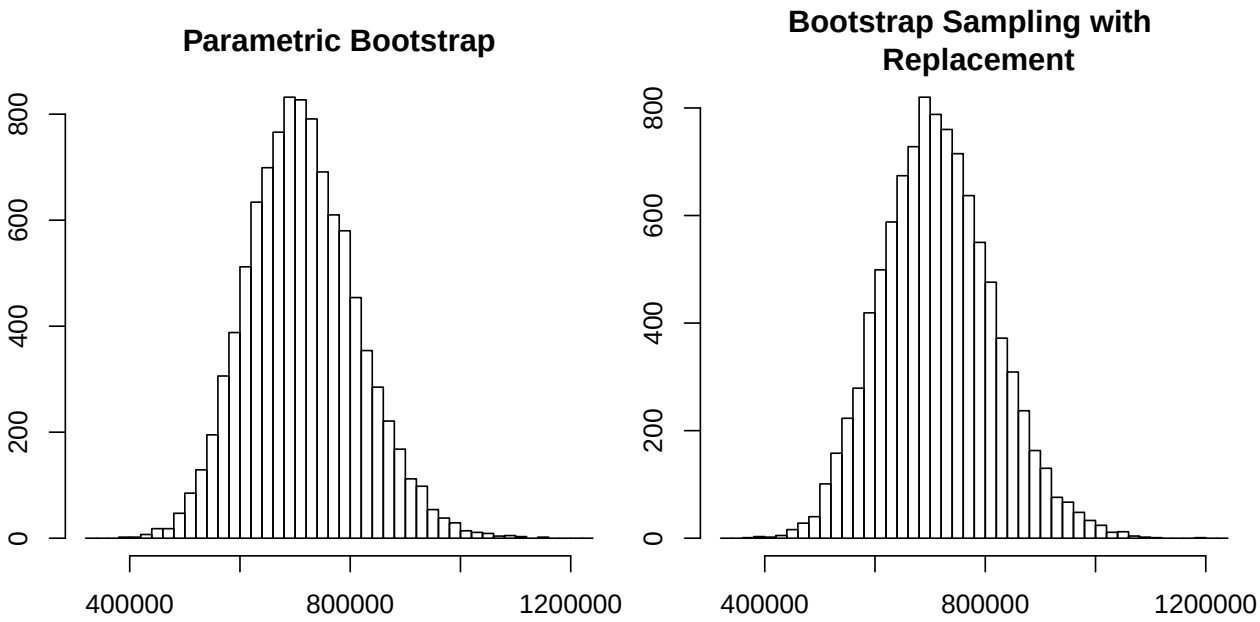
```
## [1] 101643.7
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  369040  644437  709961  714699  780376 1183734
```

Comparing parametric and non-parametric results

- We can quantify estimating the standard error for the average, or other attributes, using the parametric and non-parametric bootstrap.

```
pseq = seq(0, 1, length.out=1000)
par(mfrow=c(1,2), mar=2.5*c(1,1,1,0.1))

hhPopAve <- hist(extendrange(c(bootAvg,bootAvg0)), breaks = 50, plot = FALSE)$breaks

hist(bootAvg, main="Parametric Bootstrap", breaks= hhPopAve)
hist(bootAvg0, main="Bootstrap Sampling with \n Replacement", breaks= hhPopAve)
```

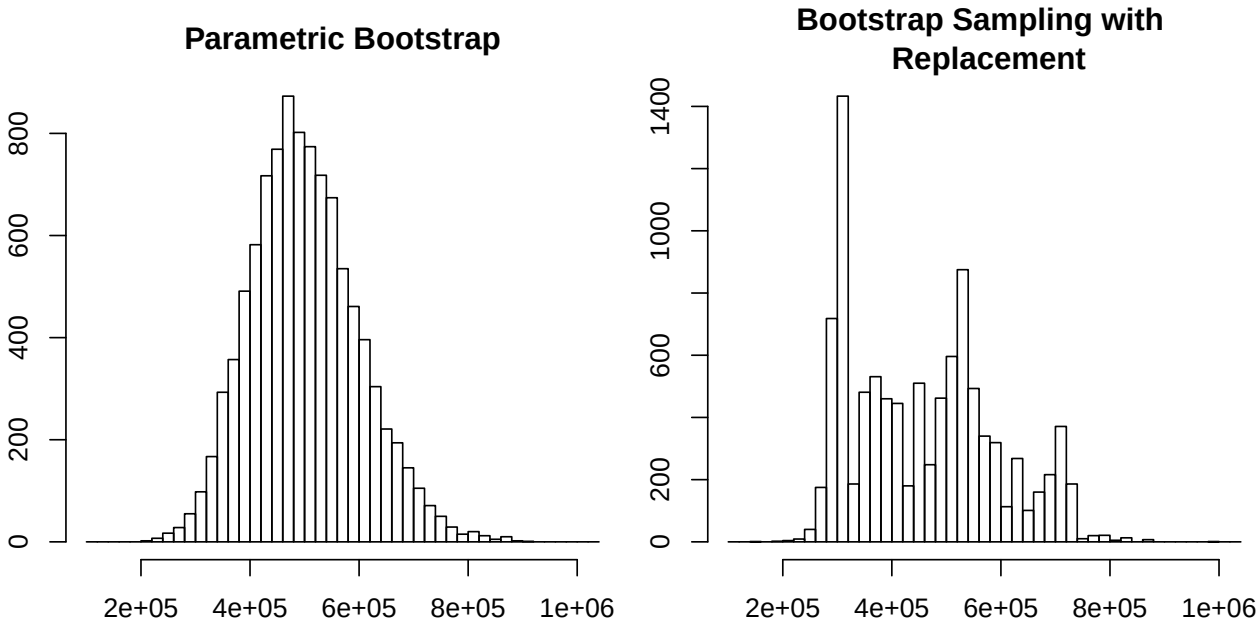


- The histograms show very similar distributions for the sample average across the two methods.
- Compare the summary statistics of the parametric and non-parametric samples to see how close the findings of two methods are.
  - This is because the exponential distribution was a good fit to the data.
  - It may not work as well for statsitics other than the sample average though (see example below)

Example - Median

- Now consider the median for the same data, i.e.
  - the West region in the US agricaulture data, and suppose we obtain a sample of size 50 from the 422 farms in Western region and measure the number of acres in 1987.

```
bootMed = apply(Sstar,2, median)
bootMed0 = apply(Sstar0,2,median)

pseq = seq(0, 1, length.out=1000)
par(mfrow=c(1,2), mar=2.5*c(1,1,1,0.1))

hhPopMed <- hist(extendrange(c(bootMed,bootMed0)), breaks = 50, plot = FALSE)$breaks

hist(bootMed, main="Parametric Bootstrap", breaks= hhPopMed)
hist(bootMed0, main="Bootstrap Sampling with \n Replacement", breaks= hhPopMed)
```



- What do you observe?
- Try other attributes like IQR, min, max, midhinge, CV, etc.

Bootstrap in Regression

- For this section, for simplicity we will treat each data-set as a sample.

Iris Data

- Here we will explore the bootstrap in analyzing the famous Iris data-set.
  - Iris is a data set with 150 cases and 5 variables named Sepal.Length, Sepal.Width, Petal.Length, Petal.Width, and Species.
  - We will limit ourselves to the Setosa flower, in which Sepal.Length is our x-covariate and the Sepal.Width is our response.

The Iris data is built-in to R and a part of it is shown below:

```
data(iris)
head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
```

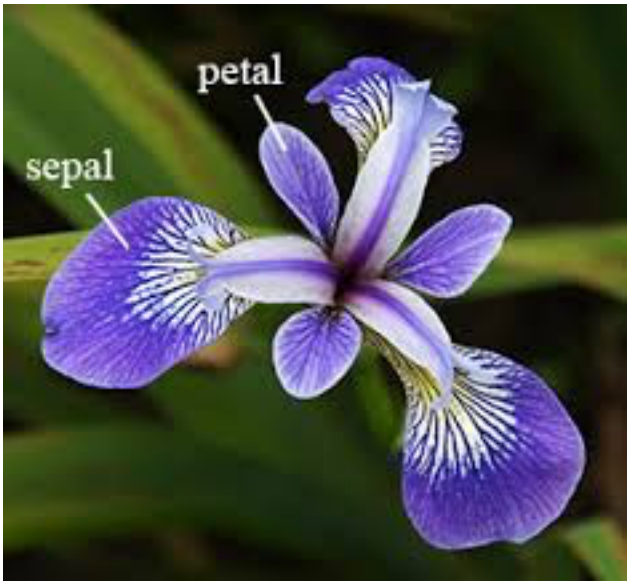


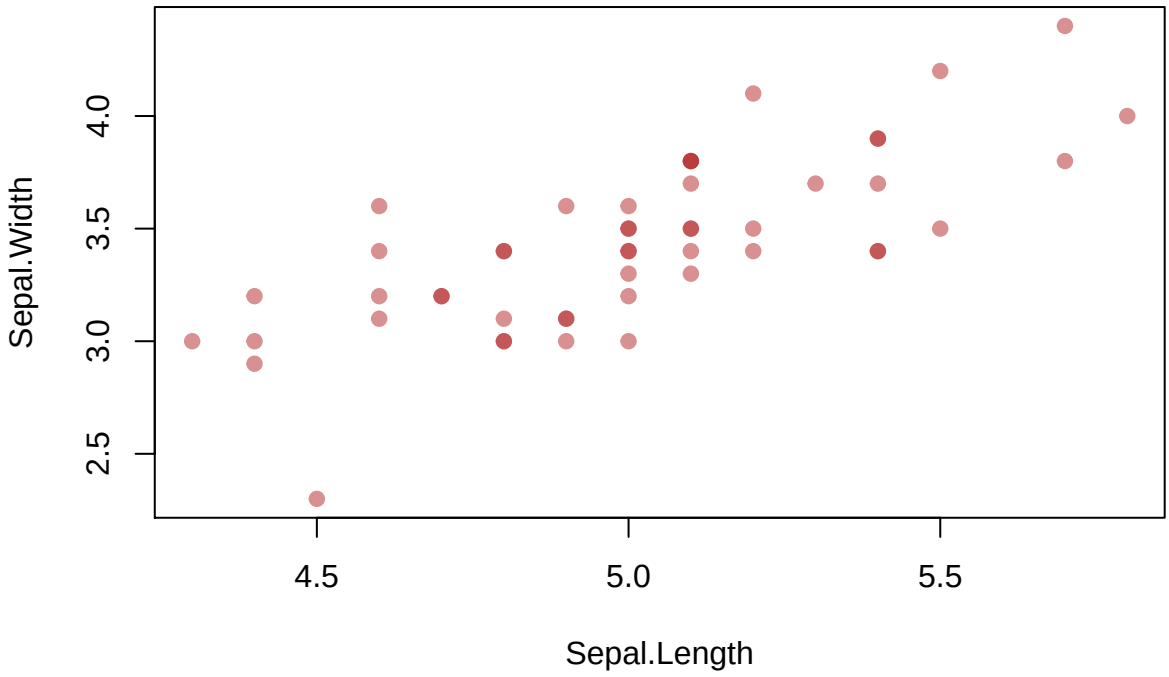
Figure 1: Iris Flower

```
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
iris.s = iris[iris[,5] == "setosa",-c(3,4,5)]
#head(iris.s)
```

The septal width and septal length seem to be correlated:

```
plot(iris.s, col=adjustcolor("firebrick", 0.5), pch=19)
```



- Here we assume that
  - sepal.Length is our x-covariate and

  - the Sepal.Width is our response.

```
x = iris.s$Sepal.Length
y = iris.s$Sepal.Width
n = length(y)
```

- The assumed regression model is

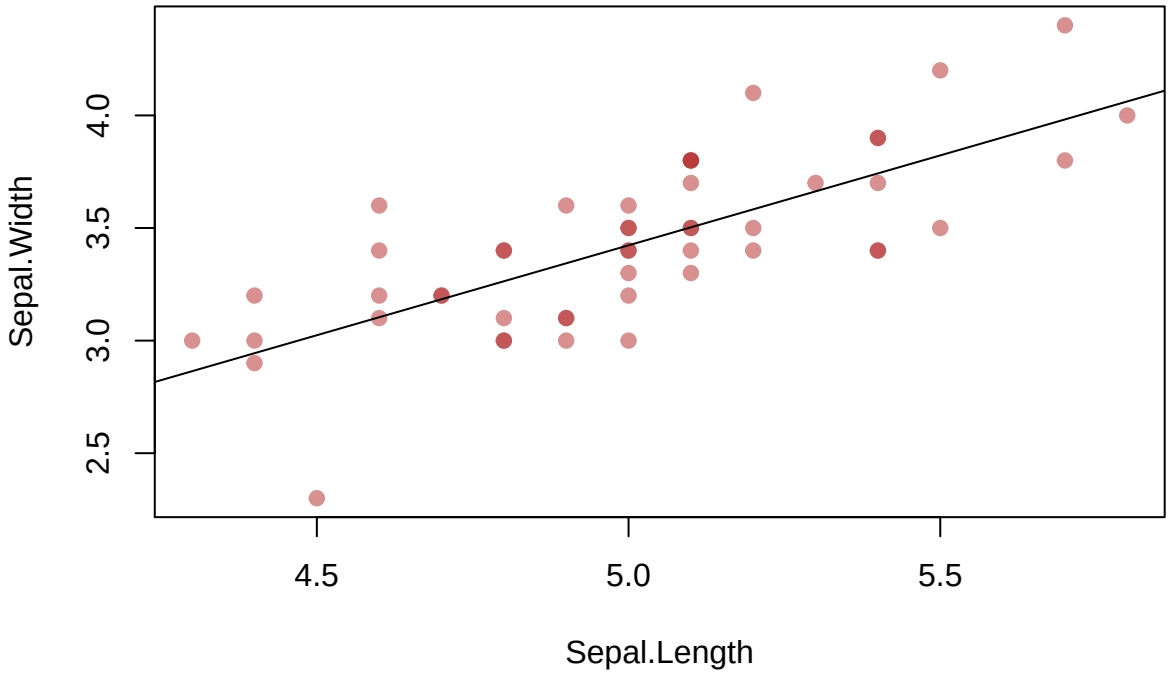
$$Y_i = \alpha + \beta\,(x_i - \overline{x}) + R_i$$

where $R_i \sim N(0, \sigma^2)$ and the least squares estimate of $(\widehat{\alpha}, \widehat{\beta})$ is

```
lm(y~x)$coef
```

```
## (Intercept)           x
##  -0.5694327   0.7985283
```

```
data(iris)
iris.s = iris[iris[,5] == "setosa",-c(3,4,5)]
#head(iris.s)
plot(iris.s, col=adjustcolor("firebrick", 0.5), pch=19)
beta.hat = lm(y~ I(x-mean(x)))$coef
abline( beta.hat +  c(-beta.hat[2]*mean(x),0) )
```



Resampling the Pairs: non-parametric bootstrap

- How can one assess the sampling variability of the regression line?
  - How can we assess the standard error of the regression line?

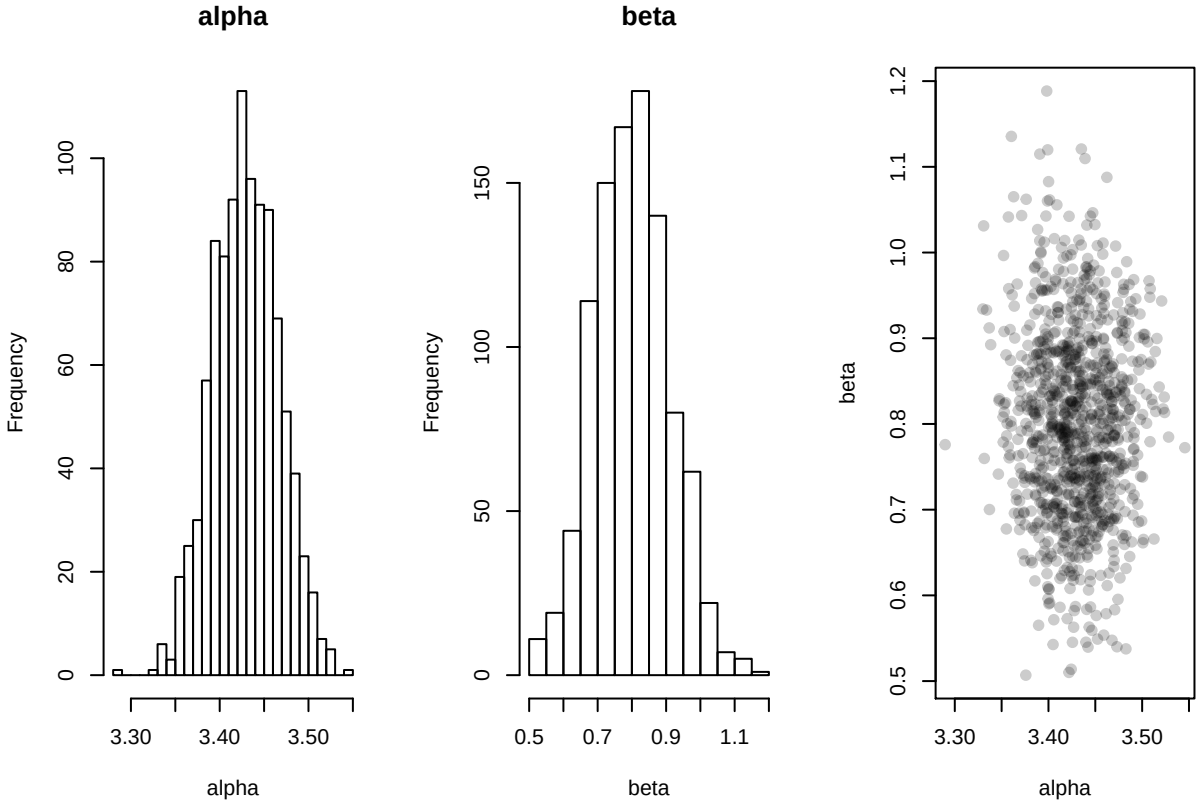
- The sample is
$$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$$
  - We sample with replacement from the pairs of observations. i.e. sample (x_i, y_i).

```
B = 1000;
beta.boot = t(sapply(1:B, FUN =function(b)
  lm(y~ I(x-mean(x)), subset=sample(n,n, replace=TRUE))$coef ))
```

- For each bootstrap sample $\mathcal{S}_b^\star$, we estimate the LS line to obtain $(\widehat{\alpha}_b^\star, \widehat{\beta}_b^\star)$, for $b = 1, \ldots, B$.

Bootstrap the Regression Coefficients

Plots of the boot estimates.



- Notice that the estimates seem to be generated from a bivariate normal (how so?)
  - which indicates that the bootstrap and the "theoretical" confidence intervals based on the errors being Gaussian should agree.

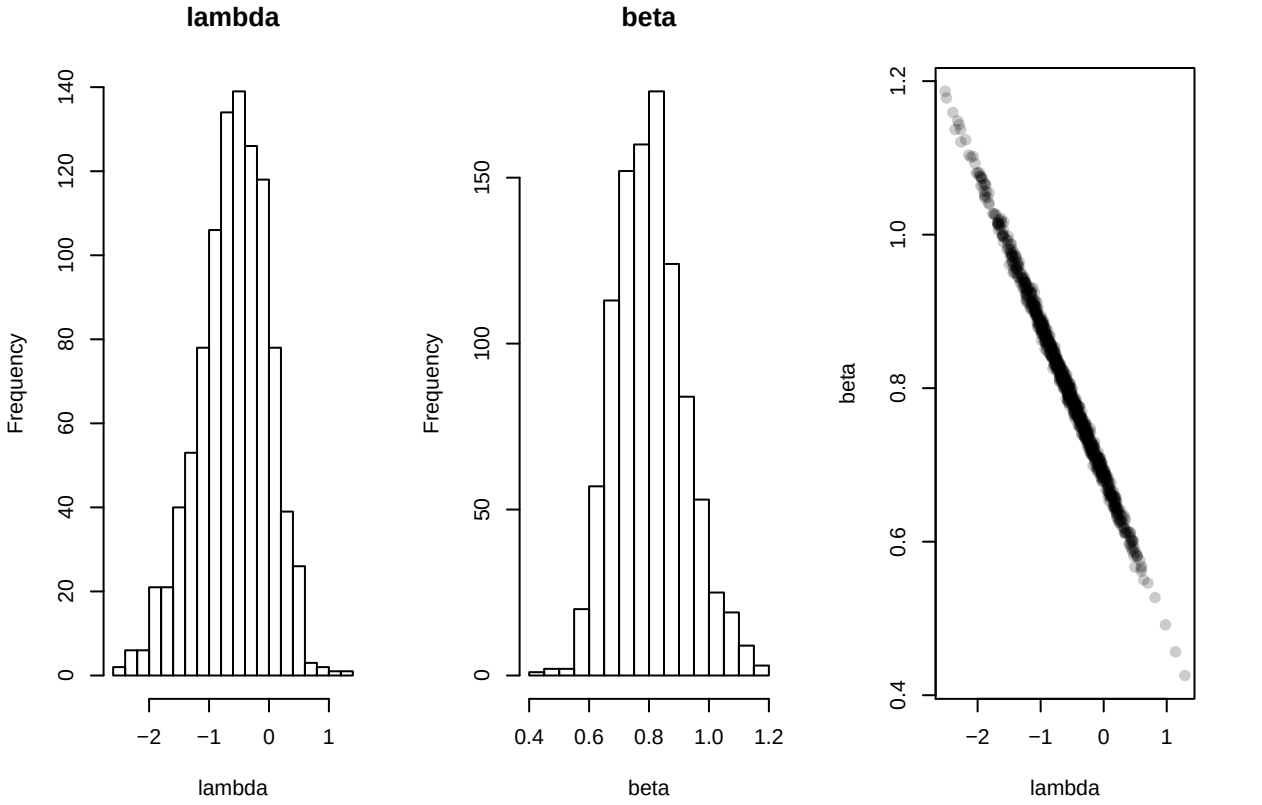
Aside: Comparing Regression Models

- Suppose we consider the alternative parameterization of the line.

$$Y_i = \alpha + \beta\,(x_i - \overline{x}) + R_i \qquad \text{and} \qquad Y_i = \lambda + \beta x_i + R_i$$

- The bootstrap replicates for this model are

```
beta.boot2 = t(sapply(1:B, FUN =function(b)
  lm(y~ x, subset=sample(n,n, replace=TRUE))$coef ))
```



- Note the strong dependence between the estimates of the two parameters (negative correlation) which does not exist when using the centralized model, i.e. $Y_i = \alpha + \beta(x_i - \overline{x}) + R_i$.

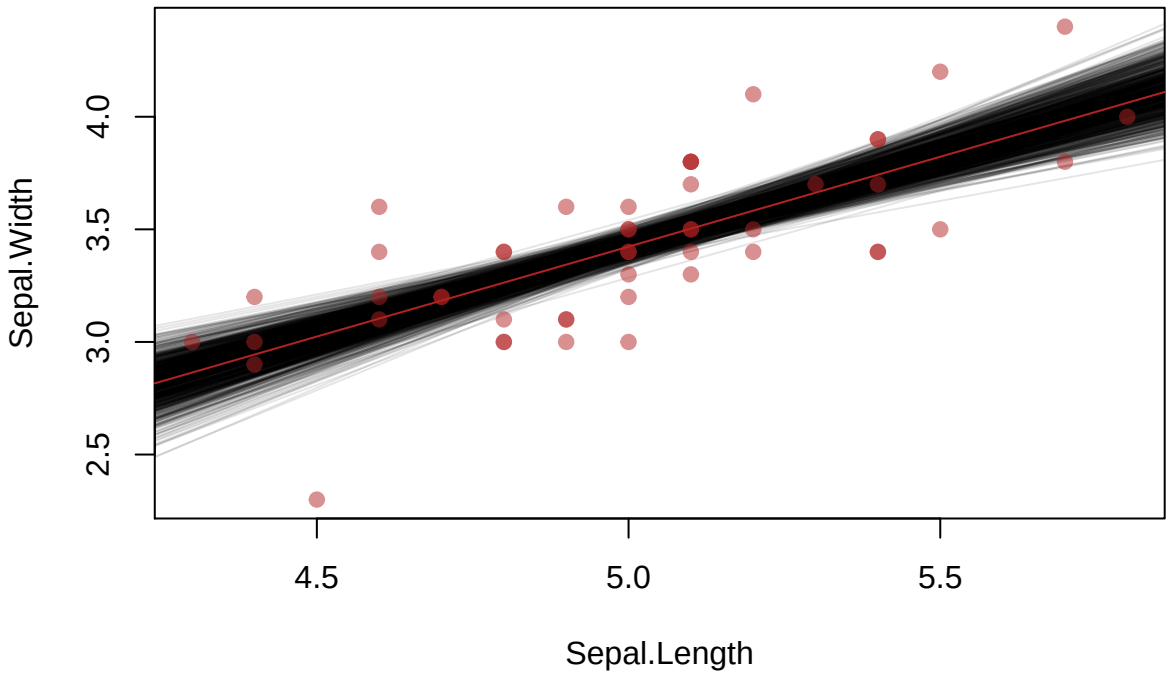
Bootstrap Regression lines

We can plot the data with the fitted model

$$y = \widehat{\alpha} + \widehat{\beta}\,(x - \overline{x})$$

and all of the bootstrap regression lines.

$$y = \widehat{\alpha}^{\star}_b + \widehat{\beta}^{\star}_b\,(x - \overline{x})$$



- How can we construct a confidence interval for the regression line?

Bootstrap Confidence Interval (Percentile Method)

Suppose we want a confidence interval for the fitted line at the value x_0

- The fitted value is

$$\widehat{\mu}(x_0) = \widehat{\alpha} + \widehat{\beta}\,(x_0 - \overline{x})$$

- The bootstrap replicates of $\widehat{y}$ are

$$\widehat{\mu}_b^\star(x_0) = \widehat{\alpha}_b^\star + \widehat{\beta}_b^\star\,(x_0 - \overline{x}) \qquad \text{where} \qquad b = 1, \ldots, B$$

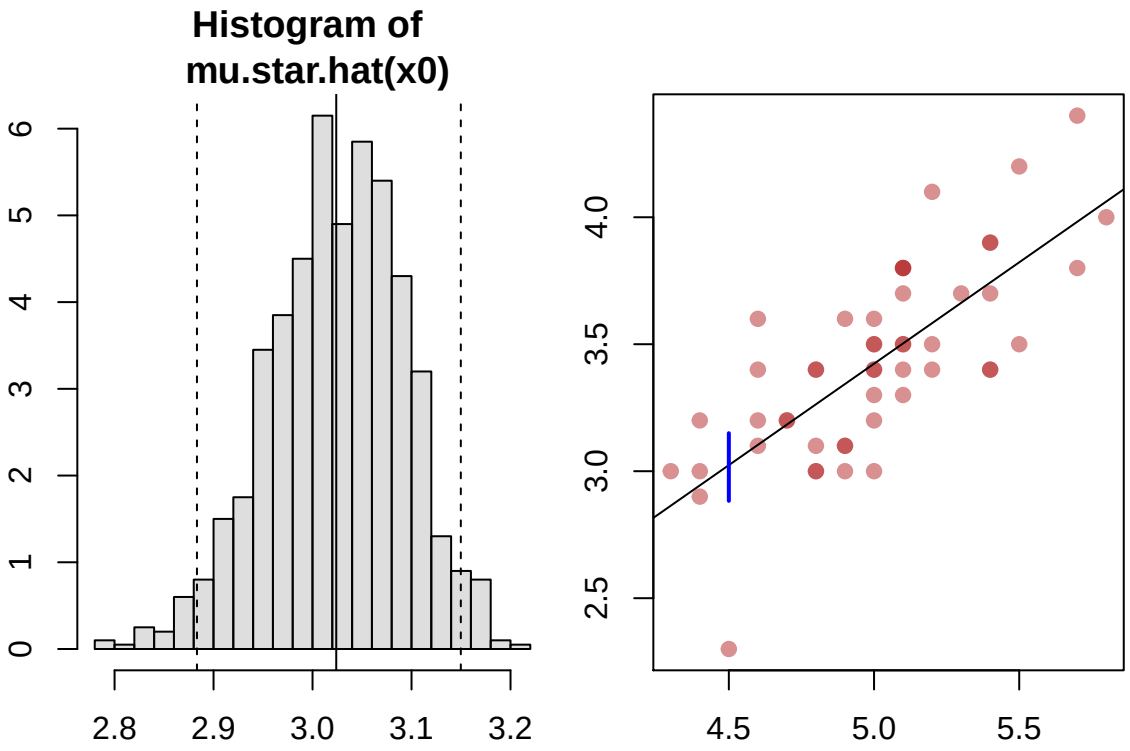
- A 95% bootstrap confidence interval is the 2.5 and 97.5 quantiles from the bootstrap replicates (percentile interval)

$$\widehat{\mu}_{lower}(x_0) = Q_{\widehat{\mu^\star}(x_0)}(0.025) \qquad \text{and} \qquad \widehat{\mu}_{lower}(x_0) = Q_{\widehat{\mu^\star}(x_0)}(0.975)$$

```
x0      = 4.5

mu0.hat = sum(  beta.hat*c( 1, x0 -  mean(x) )   )

mu0.star.hat = apply(beta.boot, 1, function(z,a) { sum(z*a) },
                     a=c( 1, x0 - mean(x) ) )

boot.ci0 = quantile( mu0.star.hat, prob= c(.025, .975))
```

- Using $x_0 = 4.5$ the fitted value, $\widehat{\mu}(x_0)$, is 3.02 and bootstrap confidence using the percentile method is 2.88, 3.15

```
par(mfrow=c(1,2), mar=2.5*c(1,1,1,0.1))

hist(mu0.star.hat, freq=FALSE, breaks="FD",
     col=adjustcolor("grey", 0.5),
     main="Histogram of \n mu.star.hat(x0)")
abline( v=c(mu0.hat, boot.ci0), lty=c(1,2,2) )

plot(iris.s, pch=19, col=adjustcolor("firebrick", 0.5))
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0) )
lines( c(x0,x0), boot.ci0, col=4, lwd=2   )
```



- To obtain a confidence interval for the whole regression line.
  - We vary x_0, perform the bootstrap again and
  - connect the lower and upper confidence intervals

```
x.seq   = c( 4.5, 5.0, 5.6 )
```

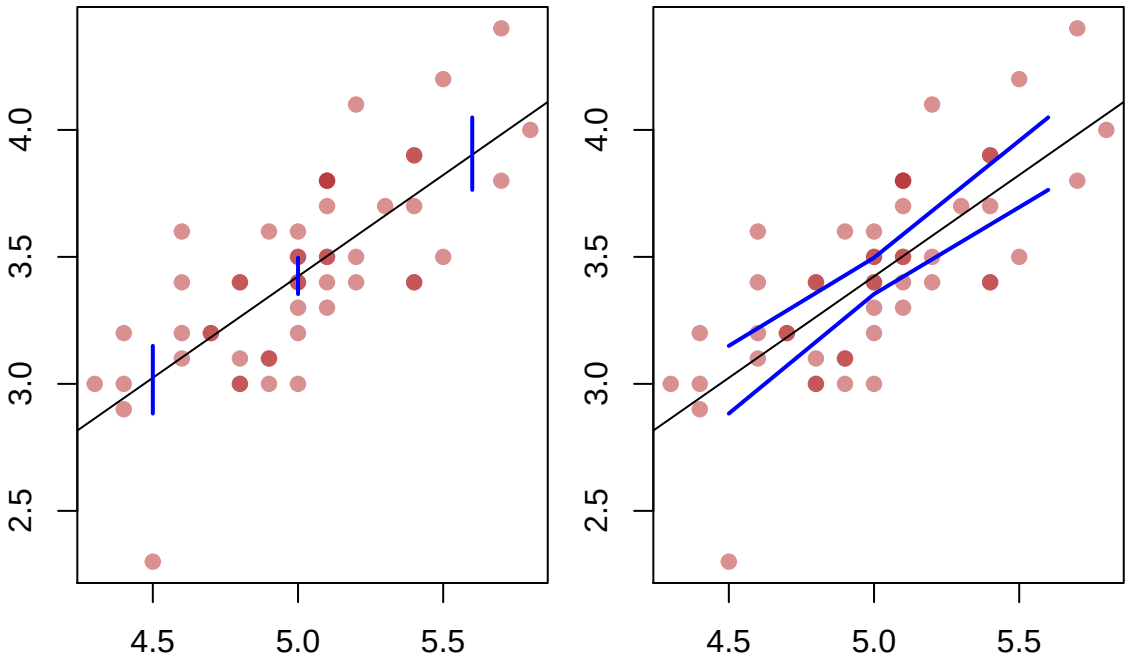
Using $x_0 = 4.5, 5, 5.6$, we obtain the following confidence intervals.

```
boot.ci = matrix(0, nrow=length(x.seq), 2)

for (i in 1:length(x.seq)) {
    y.hat = apply(beta.boot, 1, function(z,a) { sum(z*a) }, a=c( 1, x.seq[i] - mean(x) ) )
```

```
    boot.ci[i,] = quantile( y.hat, prob= c(.025, .975))
}

round(boot.ci,2)
```

```
##      [,1] [,2]
## [1,] 2.88 3.15
## [2,] 3.35 3.50
## [3,] 3.76 4.05
```

```
par(mfrow=c(1,2), mar=2.5*c(1,1,1,0.1))

plot(iris.s, pch=19, col=adjustcolor("firebrick", 0.5))
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0) )

for (i in 1:length(x.seq)) lines( rep(x.seq[i],2), boot.ci[i,], col=4, lwd=2 )

plot(iris.s, pch=19, col=adjustcolor("firebrick", 0.5))
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0) )

lines( x.seq, boot.ci[,1], col=4, lwd=2 )
lines( x.seq, boot.ci[,2], col=4, lwd=2 )
```

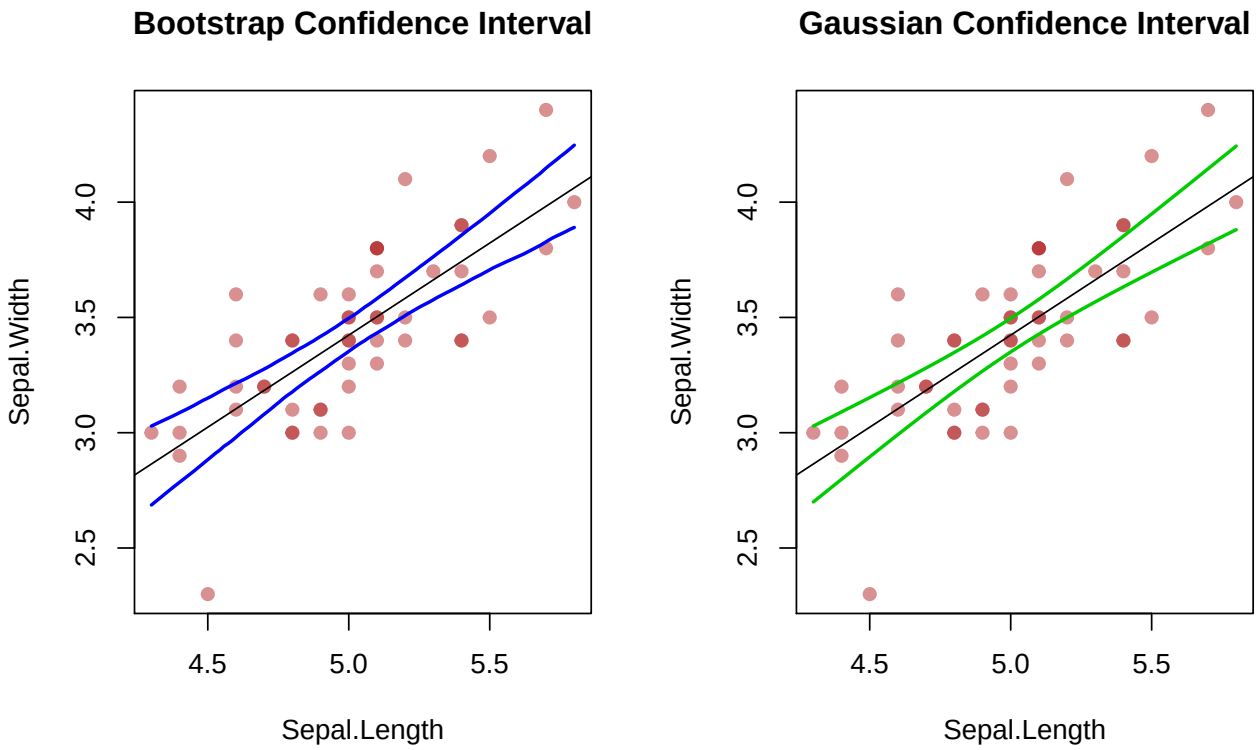


- We can add more x_0 values and then compare the bootstrap percentile interval to a confidence interval that uses the assumption of Gaussian errors.

```
x.seq   = seq( min(x), max(x), length.out=100)
boot.ci = matrix(0, nrow=length(x.seq), 2)
```

```r
for (i in 1:length(x.seq)) {
    y.hat = apply(beta.boot, 1, function(z,a) { sum(z*a) }, a=c( 1, x.seq[i] - mean(x) ) )
    boot.ci[i,] = quantile( y.hat, prob= c(.025, .975))
}

## A CI using the assumption of Gaussian Errors
ci = predict( lm(y~x), newdata=data.frame(x=x.seq), interval="confidence")
```

```r
par(mfrow=c(1,2))

plot(iris.s, pch=19, col=adjustcolor("firebrick", 0.5),
     main="Bootstrap Confidence Interval")
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0) )
lines(x.seq, boot.ci[,1], col=4, lwd=2)
lines(x.seq, boot.ci[,2], col=4, lwd=2)

plot(iris.s, pch=19, col=adjustcolor("firebrick", 0.5),
     main="Gaussian Confidence Interval")
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0) )
lines(x.seq, ci[,2], col=3, lwd=2)
lines(x.seq, ci[,3], col=3, lwd=2)
```



- A confidence interval using the assumption of Gaussian Errors and a confidence interval using the bootstrap and the percentile method match.
- Important question: what does the coverage probability of 95% for a regression line mean?
  - note that a proper definition of a regression line is $E(Y \mid X = x) = \alpha + \beta x$

Parametric Bootstrap

- How would we apply the parametric bootstrap in the context of regression?
- The assumed regression model is

$$Y_i = \alpha + \beta\,(x_i - \overline{x}) + R_i$$

  with $R_i \sim_{i.i.d} G(0, \sigma)$

- We fit the model to obtain the estimates $\widehat{\alpha}$, $\widehat{\beta}$, and $\widehat{\sigma}$
  - For thr Iris data above, these are $\widehat{\alpha} = 3.428$, $\widehat{\beta} = 0.7985$, and $\widehat{\sigma} = 0.2565$.
  - Hence the fitted model is $Y_i = 3.428 + 0.7985\,(x_i - 5.006) + R_i$ with $R_i \sim_{i.i.d} G(0, 0.2565)$

- To obtain a bootstrap samples, we generate $R_i^\star$ from $G(0, \widehat{\sigma})$ and set

$$y_i^\star = \widehat{\alpha} + \widehat{\beta}\,(x_i - \overline{x}) + R_i^\star$$

  and then the bootstrap sample is

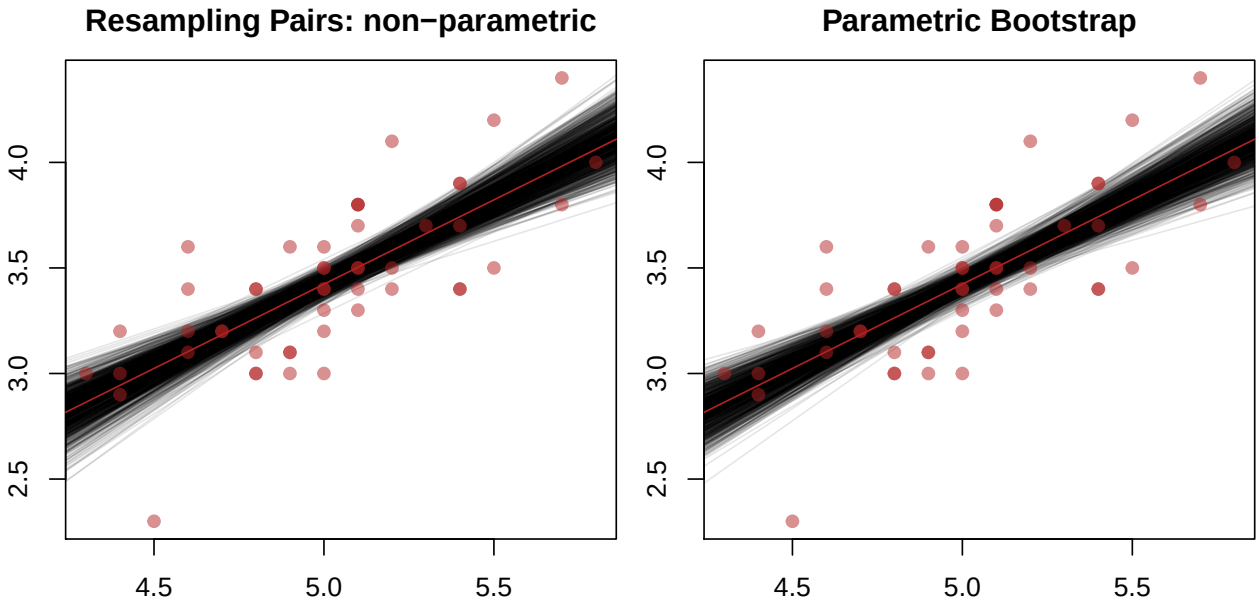
$$S_b^\star = \{(x_1, y_1^\star), (x_2, y_2^\star), \ldots, (x_n, y_n^\star)\}$$

- For each bootstrap sample $S_b^\star$ we estimate the parameters to get the bootstrap replicates $\widehat{\alpha}_b^\star$, $\widehat{\beta}_b^\star$, and $\widehat{\sigma}_b^\star$

Illustration

- Obtain the bootstrap samples using $\widehat{\alpha} = 3.428$, $\widehat{\beta} = 0.7985$, and $\widehat{\sigma} = 0.2565$.

```
B = 1000;
par.boot.sam = Map(function(b)
  { Rstar = rnorm(n, mean=0, sd= 0.2565)
    y = 3.428 + 0.7985*(x - mean(x)) + Rstar
    data.frame( x = x, y= y )  } , 1:B)

par.boot.coef = Map(function(sam)
  lm(y~I(x-mean(x)), data=sam )$coef, par.boot.sam)
```



- We notice that the results are very similar in this example.
- The parametric bootstrap motivates another way to re-sample data, i.e. sampling the errors.

Resampling the Errors

- Suppose the regression model is
$$Y_i = \alpha + \beta\,(x_i - \overline{x}) + R_i$$
where $R_i \sim F$, i.e. the errors come from some unknown density F.
  - How might we estimate F?

- We fit the model to find $\widehat{\alpha}$, $\widehat{\beta}$, and $\widehat{\sigma}$
  - then obtain the residuals $\widehat{r}_i = y_i - \widehat{y}_i = y_i - \left[\widehat{\alpha} + \widehat{\beta}\,(x_i - \overline{x})\right]$ and
  - the sample of residuals or estimates of the errors is $\widehat{\mathcal{R}} = \{\widehat{r}_1, \ldots, \widehat{r}_n\}$

- We can use the sample of residuals to estimate F using the empirical cdf.
$$\widehat{F}(t) = \frac{1}{n}\sum_{i=1}^{n} I\,(\widehat{r}_i \leq t)$$

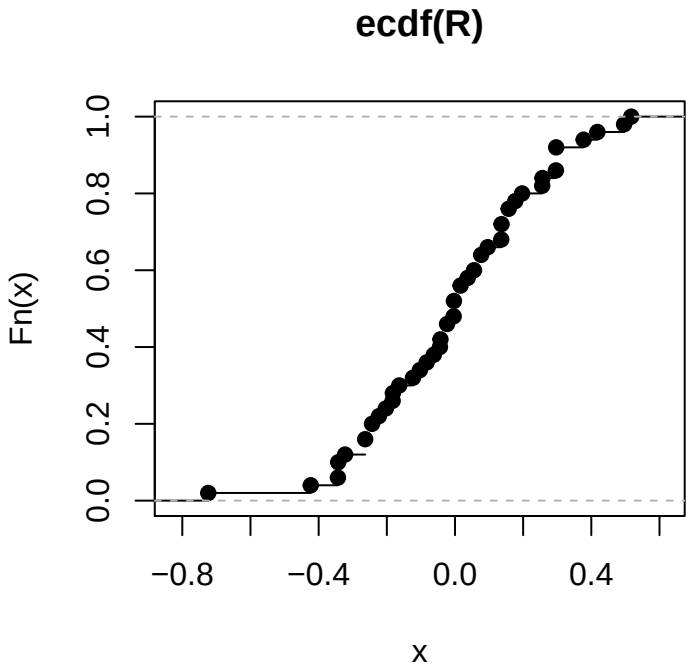
- We perform the bootstrap using $\widehat{F}$, i.e. we generate a bootstrap sample of errors $R_i^\star$ by resampling from $\widehat{\mathcal{R}}$ and obtain
$$y_i^\star = \widehat{\alpha} + \widehat{\beta}\,(x_i - \overline{x}) + R_i^\star$$

and then the bootstrap sample is

$$S_b^\star = \{(x_1, y_1^\star), (x_2, y_2^\star), \ldots, (x_n, y_n^\star)\}$$

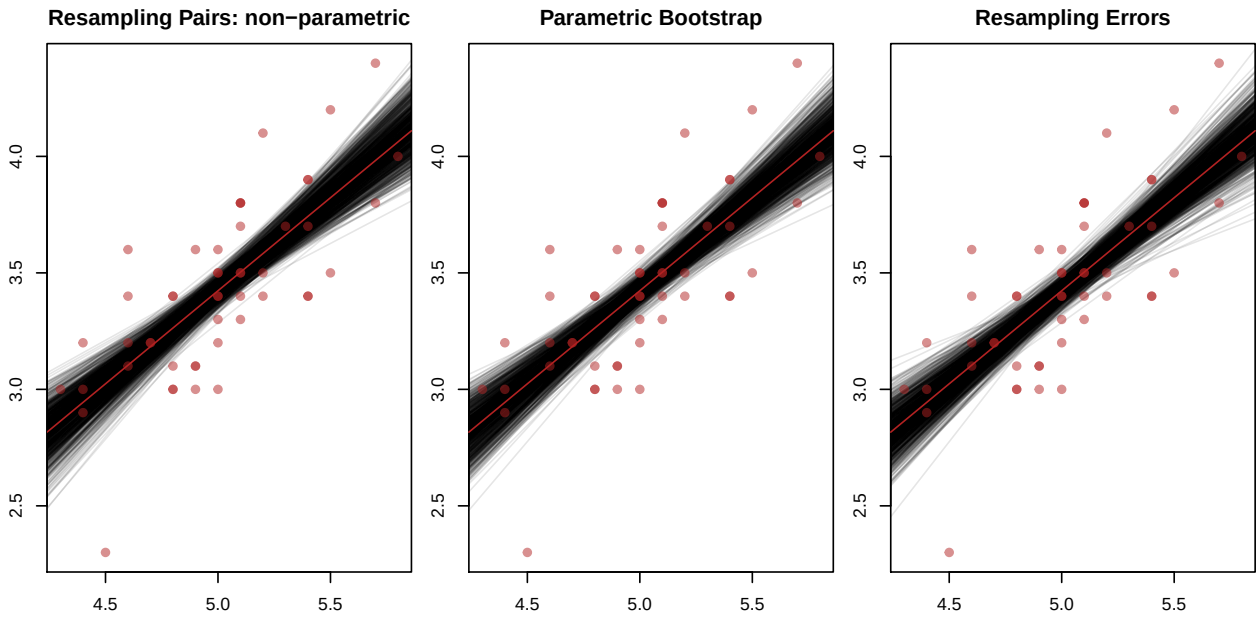
Illustration

- The sample of residuals and the empirical cdf is



- Obtain the bootstrap samples

```
B = 1000;
nonpar.boot.sam = Map(function(b)
  { Rstar = R[sample(n,n,replace=TRUE)]
    y = 3.428 + 0.7985*(x - mean(x)) + Rstar
    data.frame( x = x, y= y )  } , 1:B)

nonpar.boot.coef = Map(function(sam)
  lm(y~I(x-mean(x)), data=sam )$coef, nonpar.boot.sam)
```

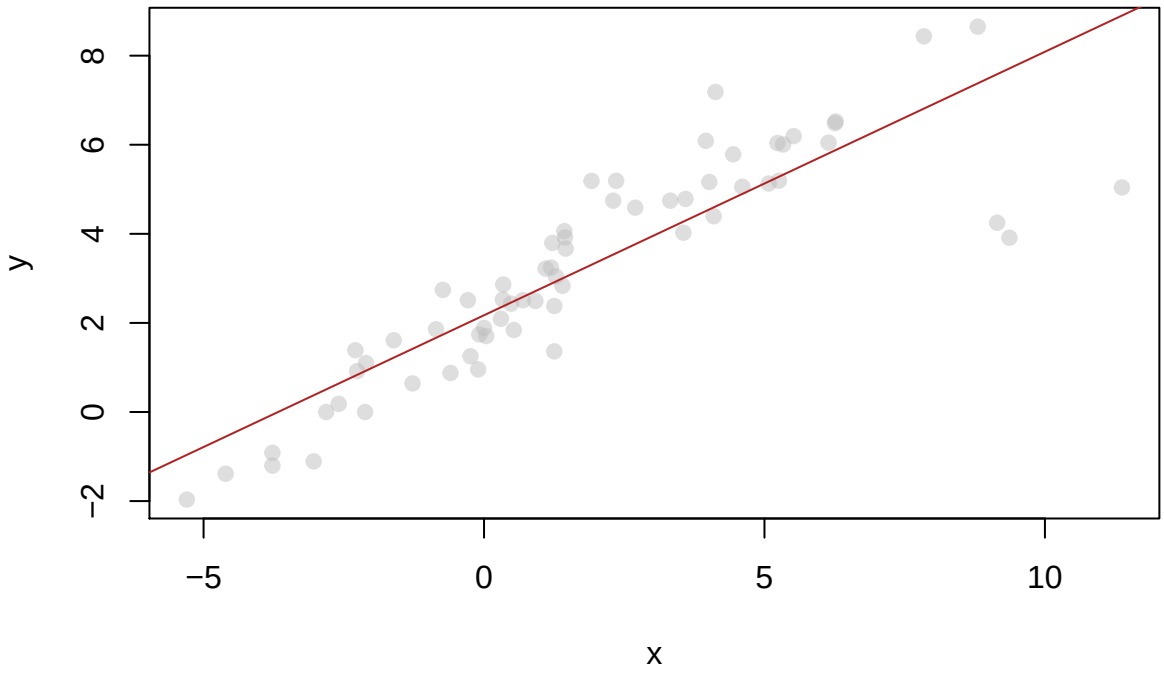


- All three methods agree

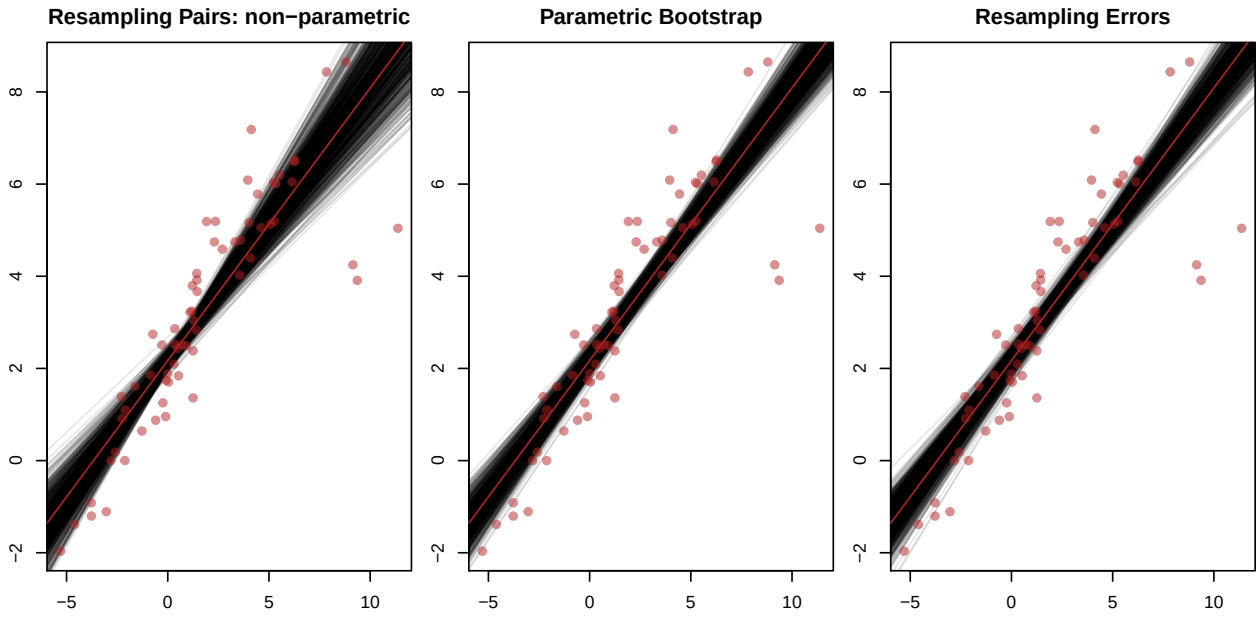
Some other Examples

Animals Data and LS

```
library(robustbase)
data(Animals2)
 #Animals2 = Animals2[-c(63,64,65),]
x = log(Animals2$body)
y  = log(Animals2$brain)
n =length(y)
plot(x,y, pch=19, col=adjustcolor("grey", .5))
beta.hat = lm(y~ I(x-mean(x)))$coef
sd.hat = sqrt(sum(lm(y~ I(x-mean(x)))$residuals^2)/(n-2) )
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0), col=adjustcolor("firebrick", 1))
```



- LS confidence Intervals

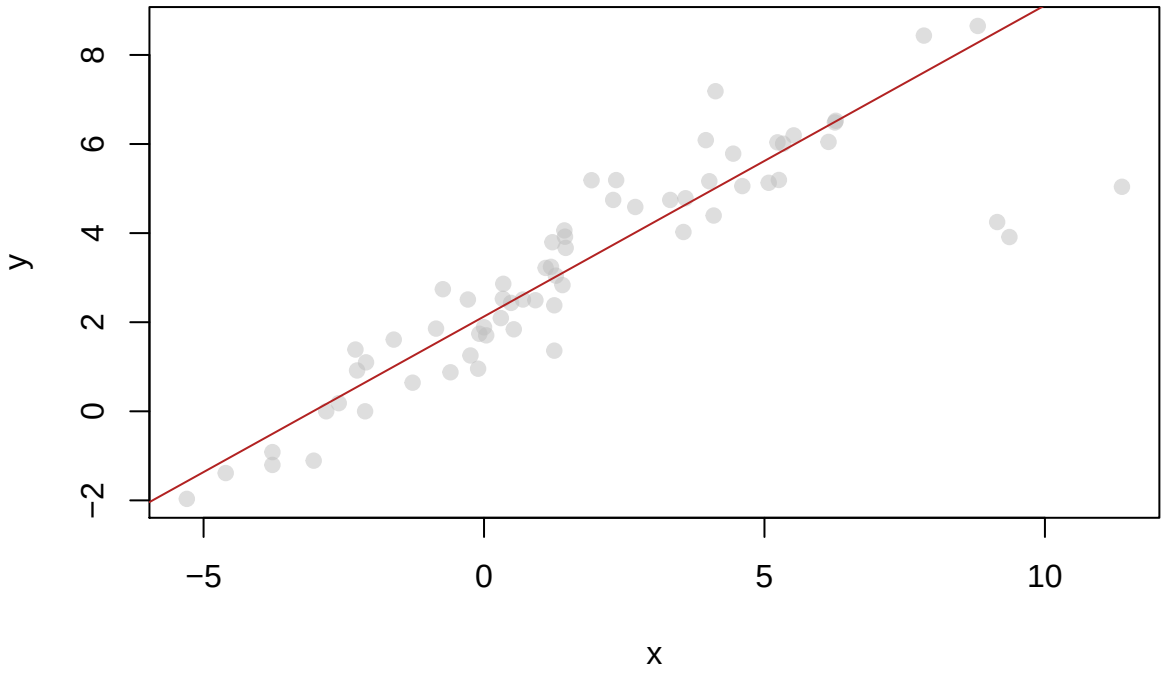


Animals Data and Robust Regression

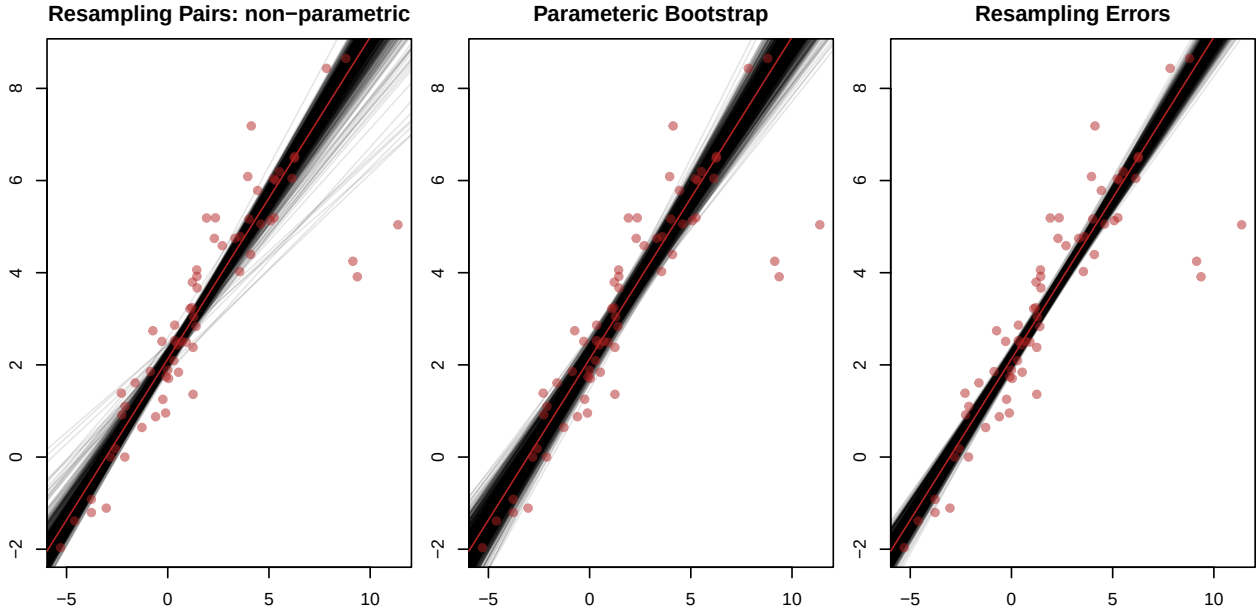
- Let fit the robust regression line using the Huber function.

```
library(robustbase)
library(MASS)
```

```r
 data(Animals2)
 #Animals2 = Animals2[-c(63,64,65),]
x = log(Animals2$body)
y  = log(Animals2$brain)
n =length(y)
plot(x,y, pch=19, col=adjustcolor("grey", .5))
beta.hat = rlm(y ~ I(x-mean(x)), psi="psi.huber")$coef

sd.hat = sqrt(sum(rlm( y~ I(x-mean(x)), psi="psi.huber")$residuals^2)/(n-2) )
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0), col=adjustcolor("firebrick", 1))
```



- Robust Regression Confidence Intervals:

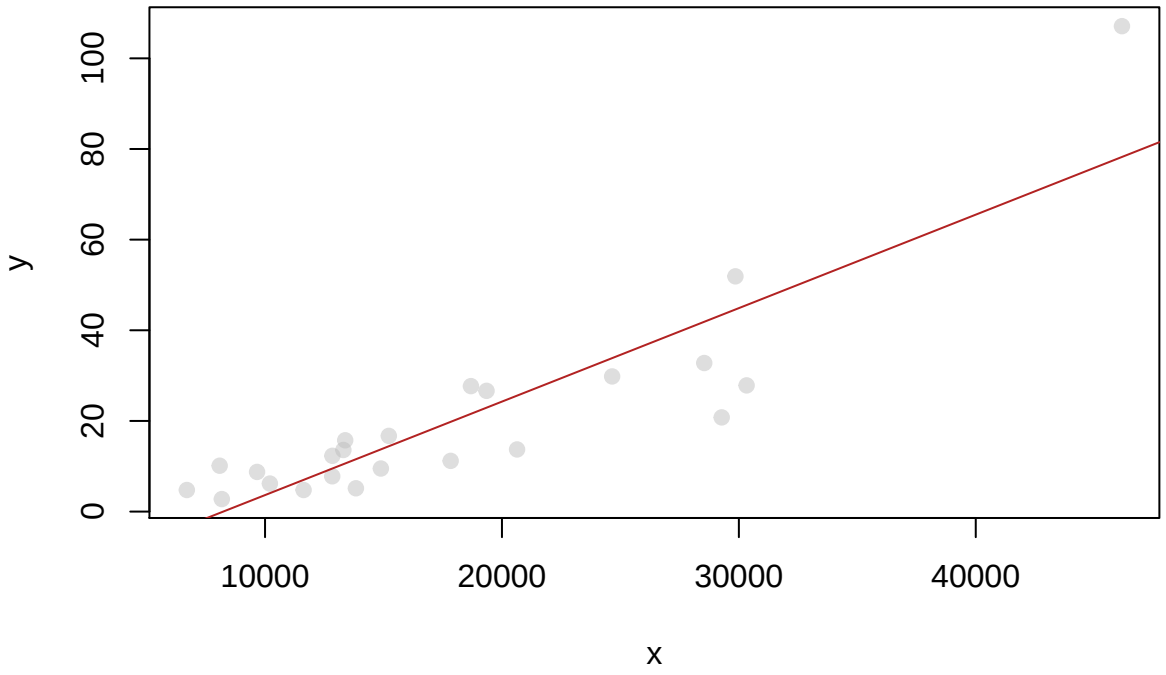


- What do you learn from these plots? Do they all agree?

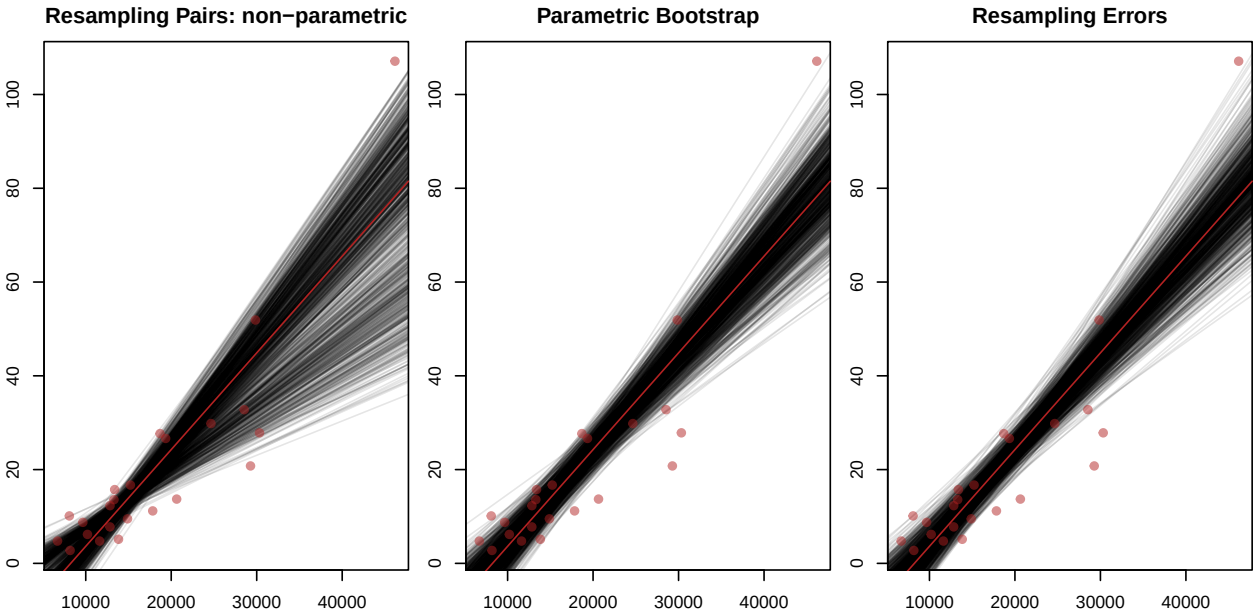
Aircraft Data and LS

- Aircraft Data deals with 23 single-engine aircraft built over the years 1947-1979, from Office of Naval Research.
  - x is the weight of the plane and
  - y is the cost in units of $\$100,000$

```
library(datasets)
x = aircraft$X3
y = aircraft$Y
n =length(y)
plot(x,y, pch=19, col=adjustcolor("grey", .5))
beta.hat = lm(y~ I(x-mean(x)))$coef
sd.hat = sqrt(sum(lm(y~ I(x-mean(x)))$residuals^2)/(n-2) )
abline(coef=beta.hat + c(-beta.hat[2]*mean(x),0), col=adjustcolor("firebrick", 1))
```



- Aircraft Data Confidence Intervals for the LS line:



- What do you learn from these plots?
- Do you think if robust regression would result in similar results?

Quote

The bootstrap "can blow the head off any problem if the statistican can stand the resulting mess."
– John Tukey