hw5.R

mwilde

Fri Nov 3 17:01:30 2017

```
# data = read.table('http://www.stat.washington.edu/tsr/s509/examples/caschool.csv',header=TRUE,sep=","

data = read.table('caschool.csv', header=TRUE, sep=",")

# see column names
names(data)
```

```
##  [1] "Observation.Number" "dist_cod"           "county"
##  [4] "district"           "gr_span"            "enrl_tot"
##  [7] "teachers"           "calw_pct"           "meal_pct"
## [10] "computer"           "testscr"            "comp_stu"
## [13] "expn_stu"           "str"                "avginc"
## [16] "el_pct"             "read_scr"           "math_scr"
```

```
summary(data$enrl_tot)
```

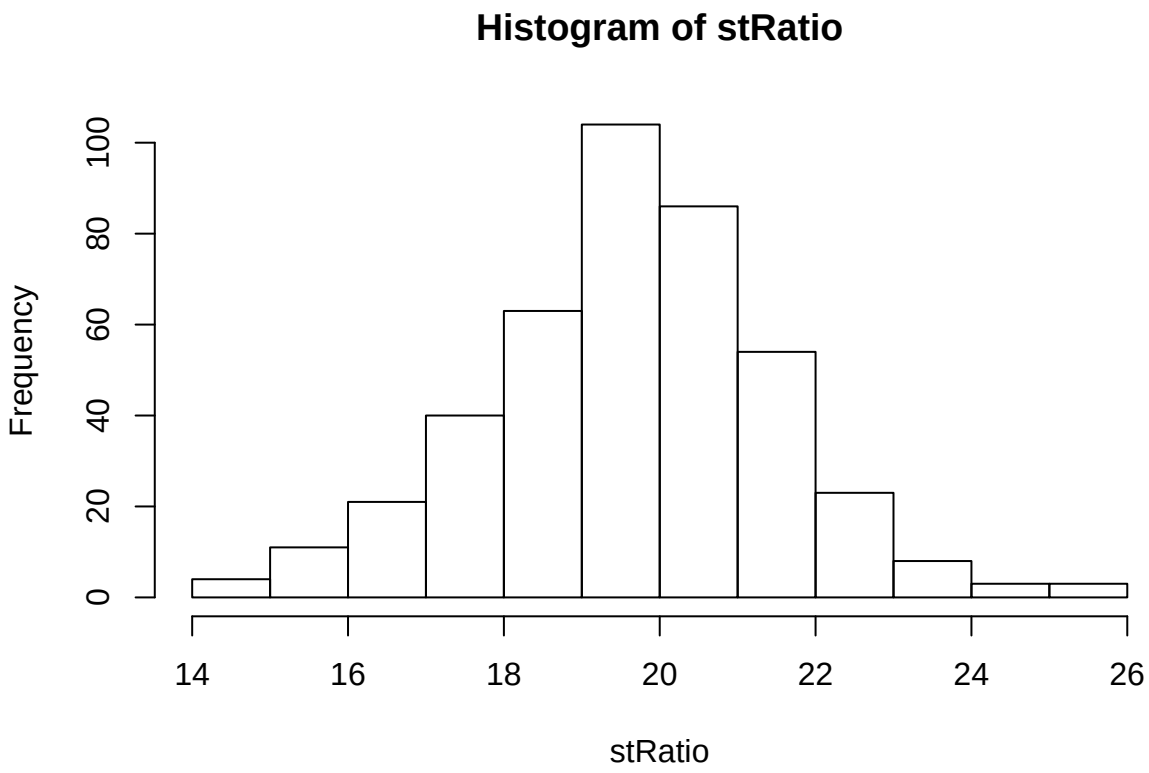
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    81.0   379.0   950.5  2628.8  3008.0 27176.0
```

```
summary(data$teachers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    4.85   19.66   48.56  129.07  146.35 1429.00
```

```
students = data$enrl_tot
teachers = data$teachers
testscr = data$testscr

# stRatio = teachers/students
stRatio = students/teachers
hist(testscr)
```


```
hist(stRatio)
```




```
plot(stRatio,testscr, xlab="Student Teacher Ratio", ylab="Test Score",
     main="Relating Student Teacher Ratio and Test Scores")
```

```r
#let's fit a linear model

fit = lm(testscr ~ stRatio)
fit # just the coefficients
```

```
##
## Call:
## lm(formula = testscr ~ stRatio)
##
## Coefficients:
## (Intercept)      stRatio
##      698.93        -2.28
```

```r
# Note that an intercept term is included by default

summary(fit) # more information on the fit
```

```
##
## Call:
## lm(formula = testscr ~ stRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.727 -14.251   0.483  12.822  48.540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 698.9330     9.4675  73.825  < 2e-16 ***
## stRatio      -2.2798     0.4798  -4.751 2.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.58 on 418 degrees of freedom
## Multiple R-squared:  0.05124,    Adjusted R-squared:  0.04897
## F-statistic: 22.58 on 1 and 418 DF,  p-value: 2.783e-06
```

```r
fit$coef # accessing the intercept and slope
```

```
## (Intercept)     stRatio
##  698.932953   -2.279808
```

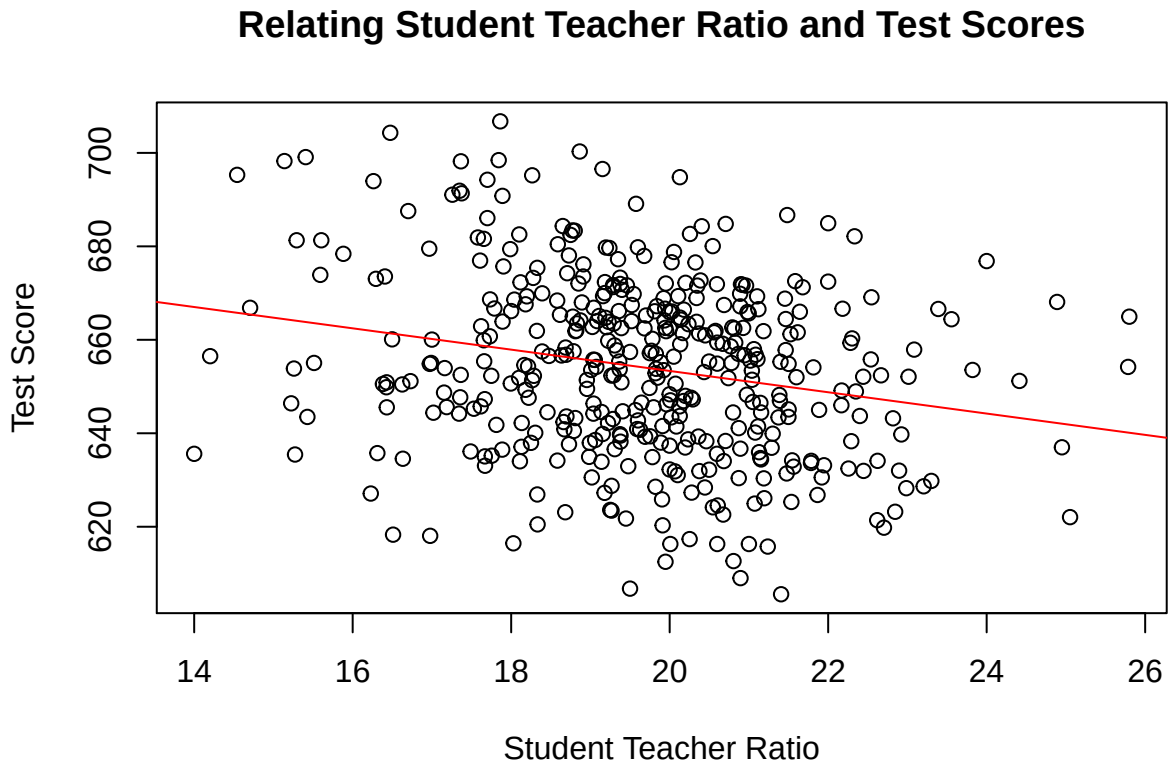
```r
cor(testscr,stRatio) #find the correlation
```

```
## [1] -0.2263628
```

```r
cor(testscr,stRatio)^2 #find the squared correlation
```

```
## [1] 0.0512401
```

```r
plot(stRatio,testscr, xlab="Student Teacher Ratio", ylab="Test Score",
     main="Relating Student Teacher Ratio and Test Scores")
abline(fit, col="red") #put the regression line on the plot
```




```
print(paste("a) The best linear predictor:"))
```

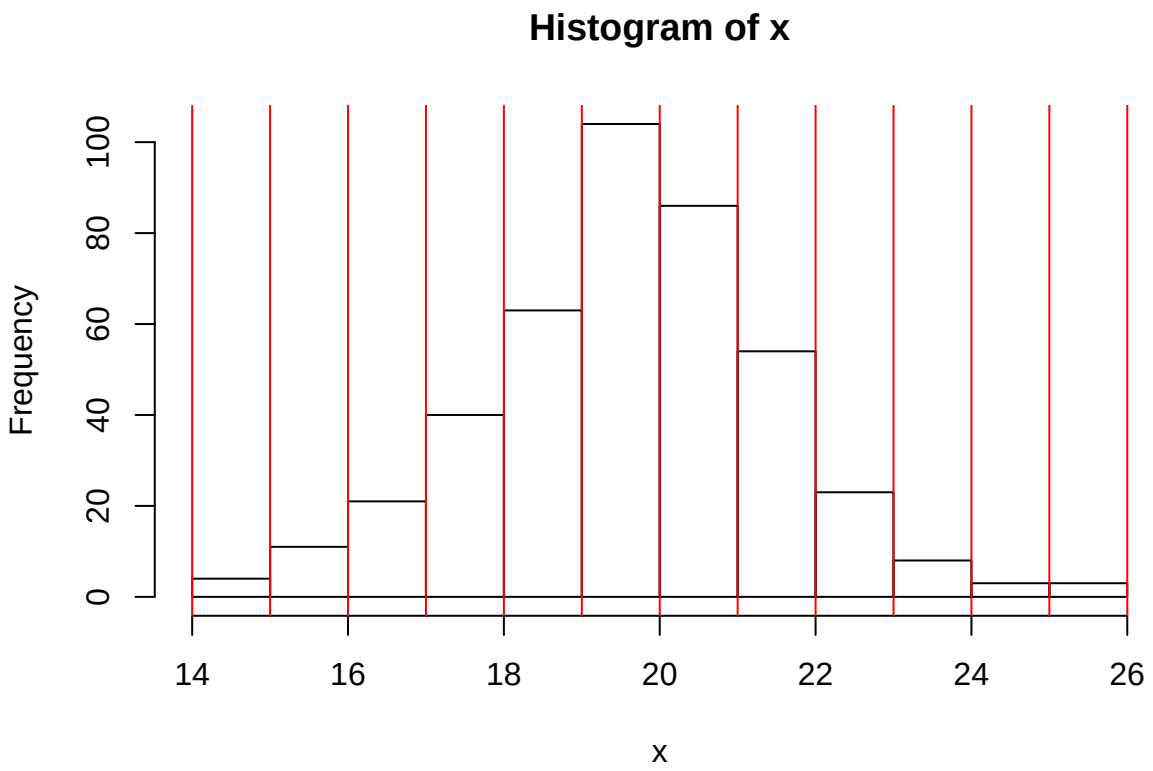
```
## [1] "a) The best linear predictor:"
```

```
print(summary(fit))
```

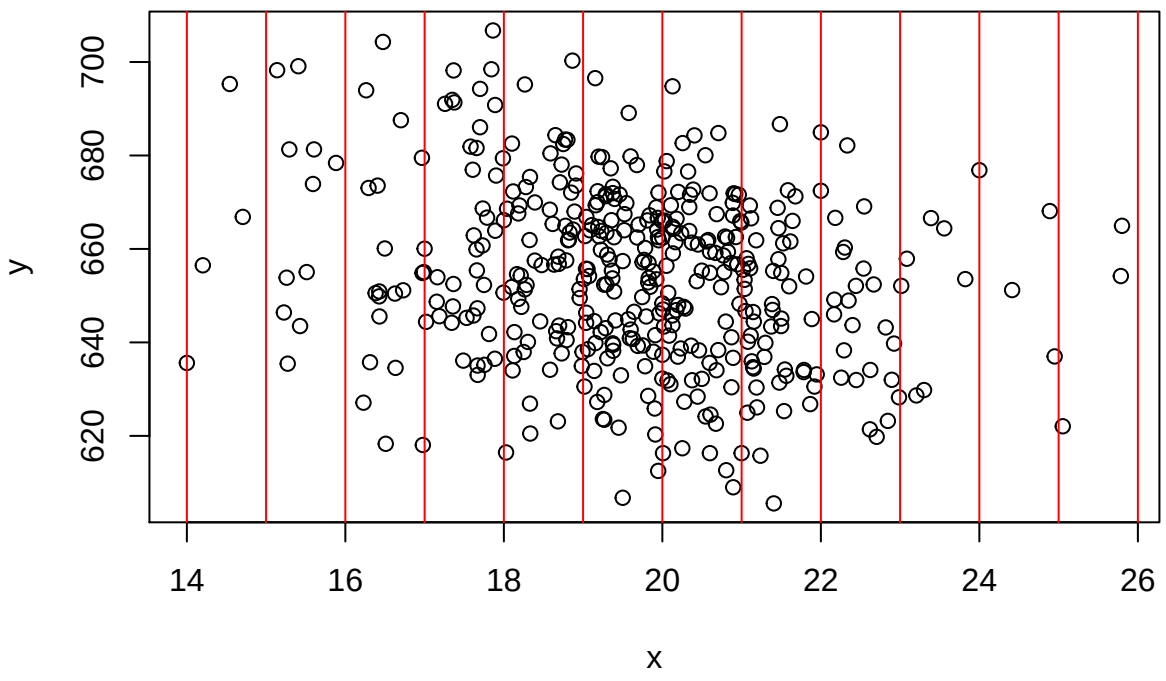
```
##
## Call:
## lm(formula = testscr ~ stRatio)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.727 -14.251   0.483  12.822  48.540
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 698.9330     9.4675  73.825  < 2e-16 ***
## stRatio      -2.2798     0.4798  -4.751 2.78e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.58 on 418 degrees of freedom
## Multiple R-squared:  0.05124,    Adjusted R-squared:  0.04897
## F-statistic: 22.58 on 1 and 418 DF,  p-value: 2.783e-06
```

```
# b) The approximate conditional expectation function
#   E[Test Score | Student Teacher Ratio] via binning
#   Student Teacher Ratio.

x = stRatio
y = testscr
```

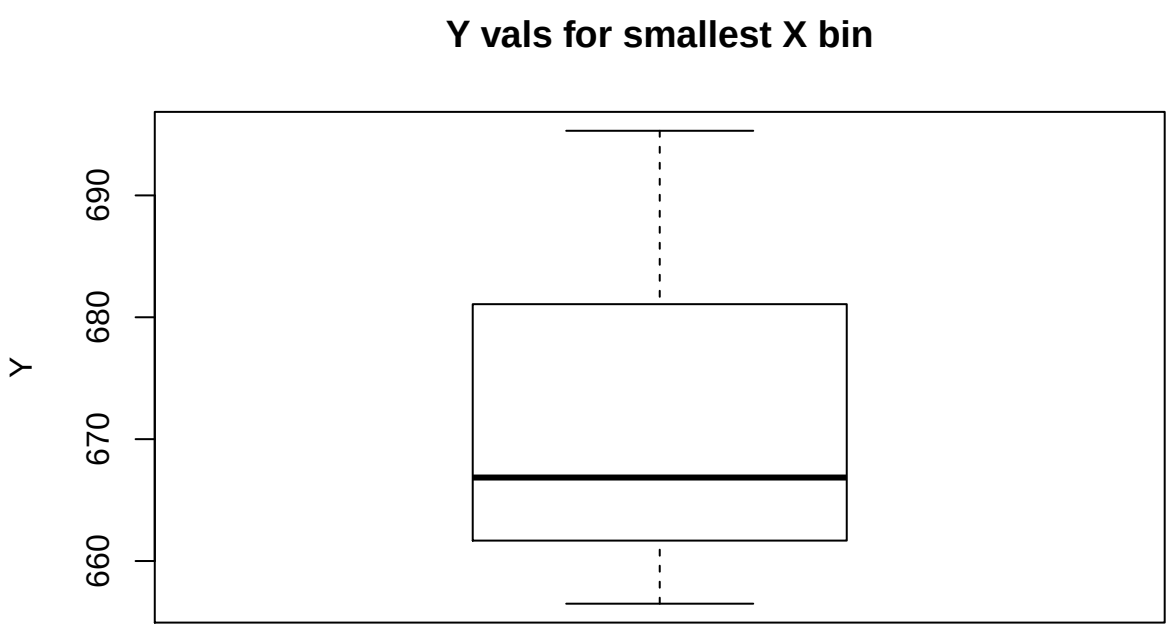
```r
hx <- hist(x)
abline(v=hx$breaks,col="red")
```



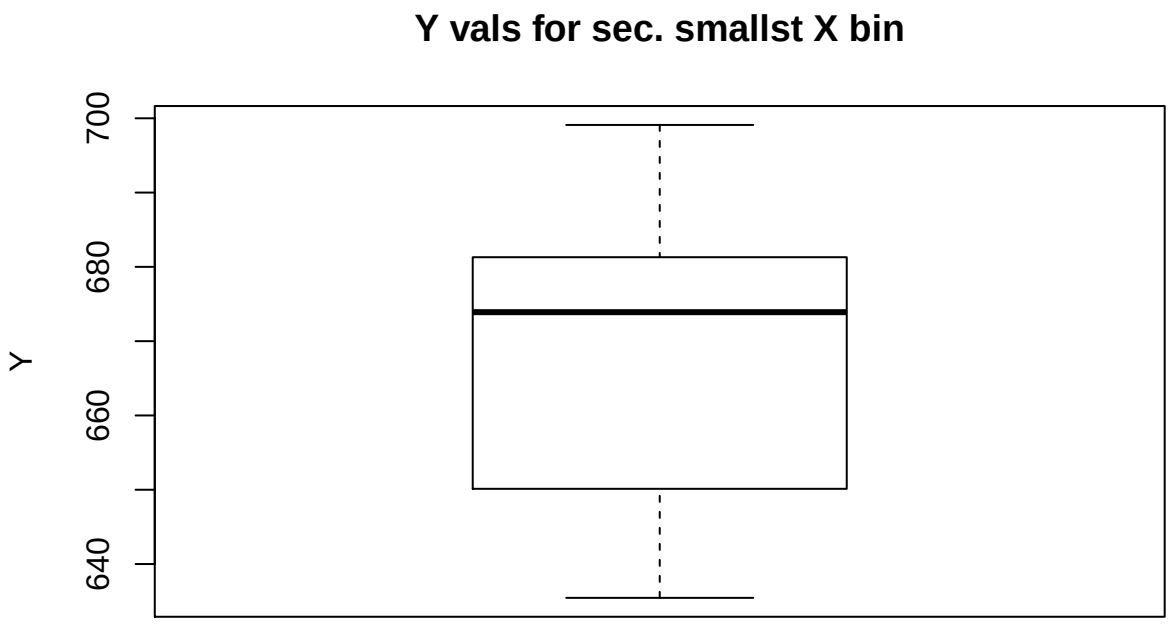

```r
plot(x,y,main="Scatterplot for data")
abline(v=hx$breaks,col="red")
```



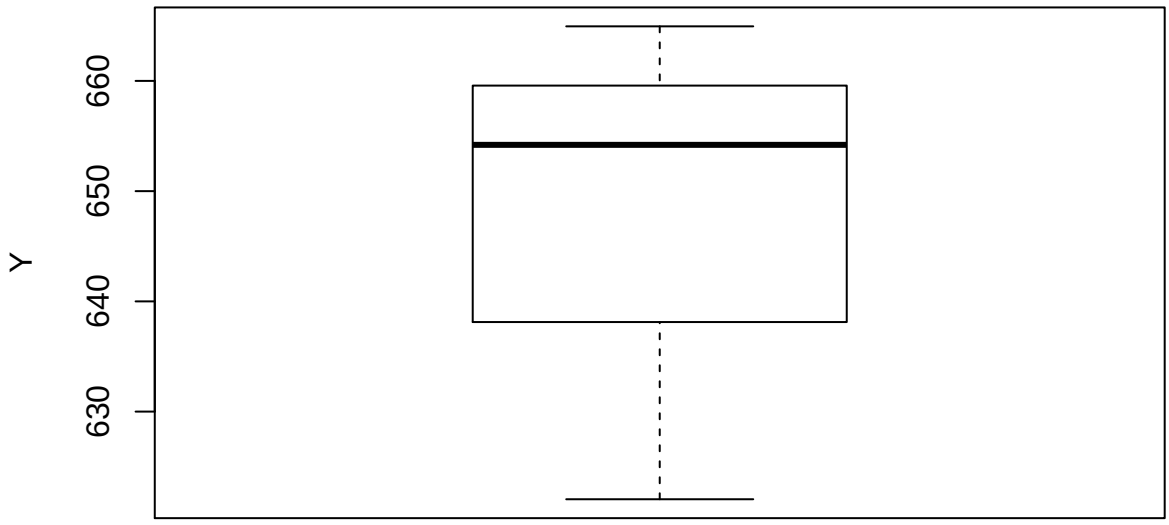
```r
z = rep(0,length(x))
bin.no = rep(0,length(x))
for (i in 1:(length(hx$breaks)-1)){
  z <- z + rep(hx$mid[i],length(x))*((x > hx$breaks[i]) & (x < hx$breaks[i+1]))
  bin.no <- bin.no + rep(i,length(x))*((x > hx$breaks[i]) & (x < hx$breaks[i+1]))
}

### Let's look at some of these subgroups
boxplot(y[z==hx$mid[1]],main="Y vals for smallest X bin",ylab="Y") # Y values for observations in small
boxplot(y[bin.no==1],main="Y vals for smallest X bin",ylab="Y") # Y values for observations in smallest
```



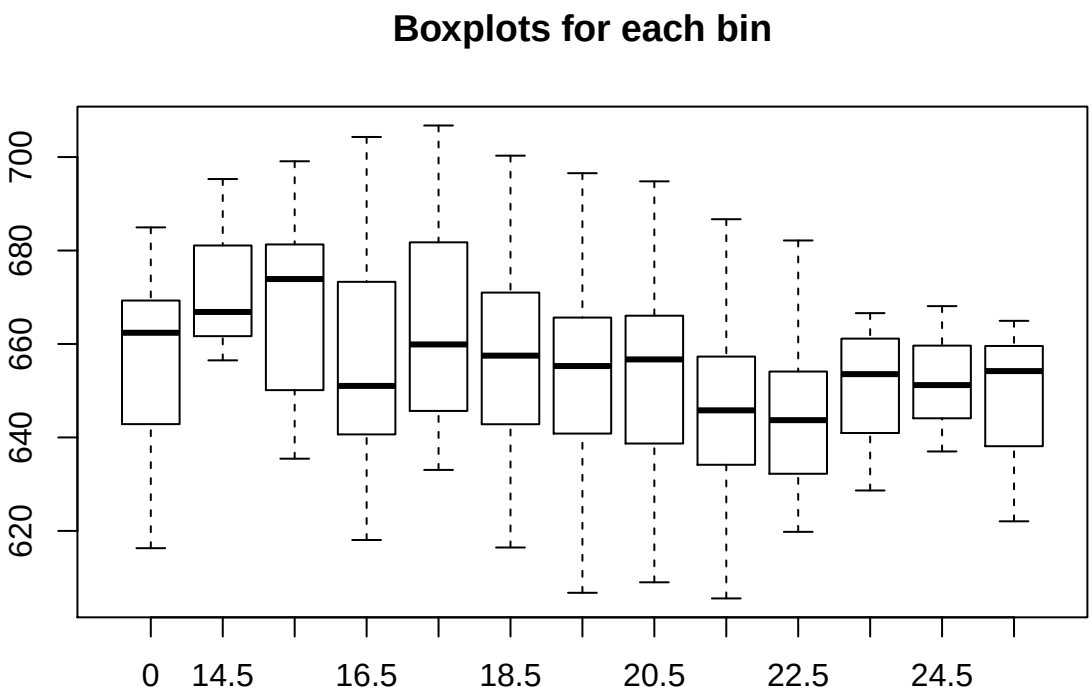

```r
boxplot(y[z==hx$mid[2]],main="Y vals for sec. smallst X bin",ylab="Y") # Y values for obs. in second sm
```



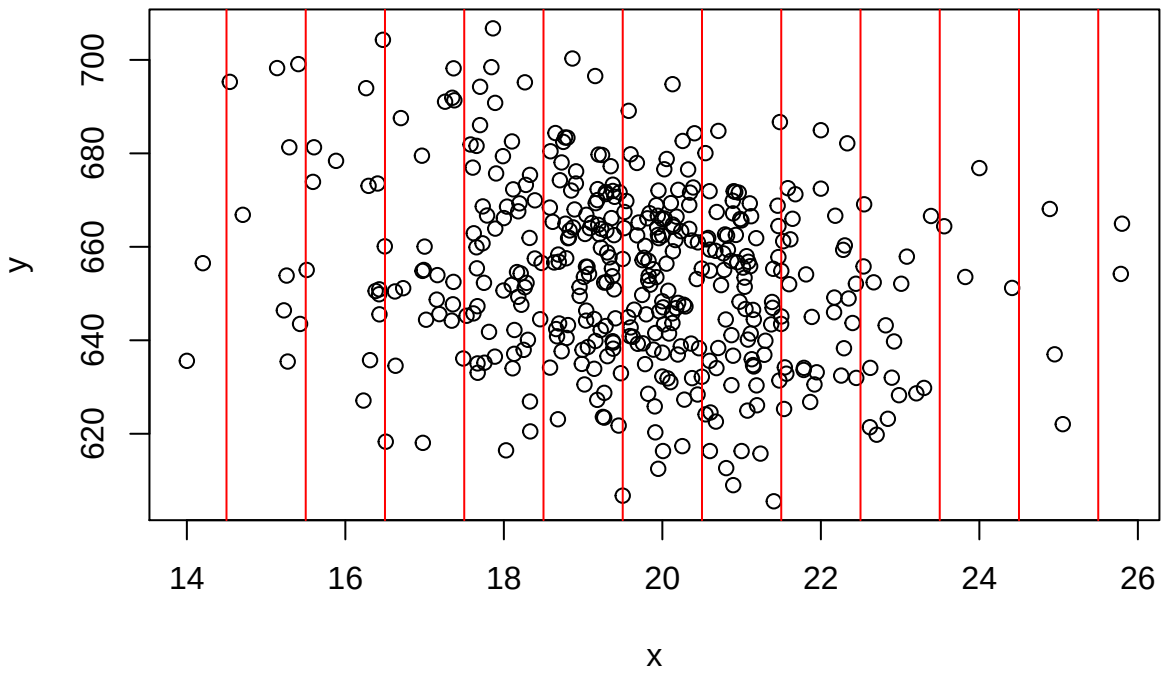

```r
boxplot(y[z==hx$mid[length(hx$breaks)-1]],main="Y vals for largest X bin",ylab="Y") # Y values for obs.
```




```
boxplot(y~z,main="Boxplots for each bin") # all at once!
```



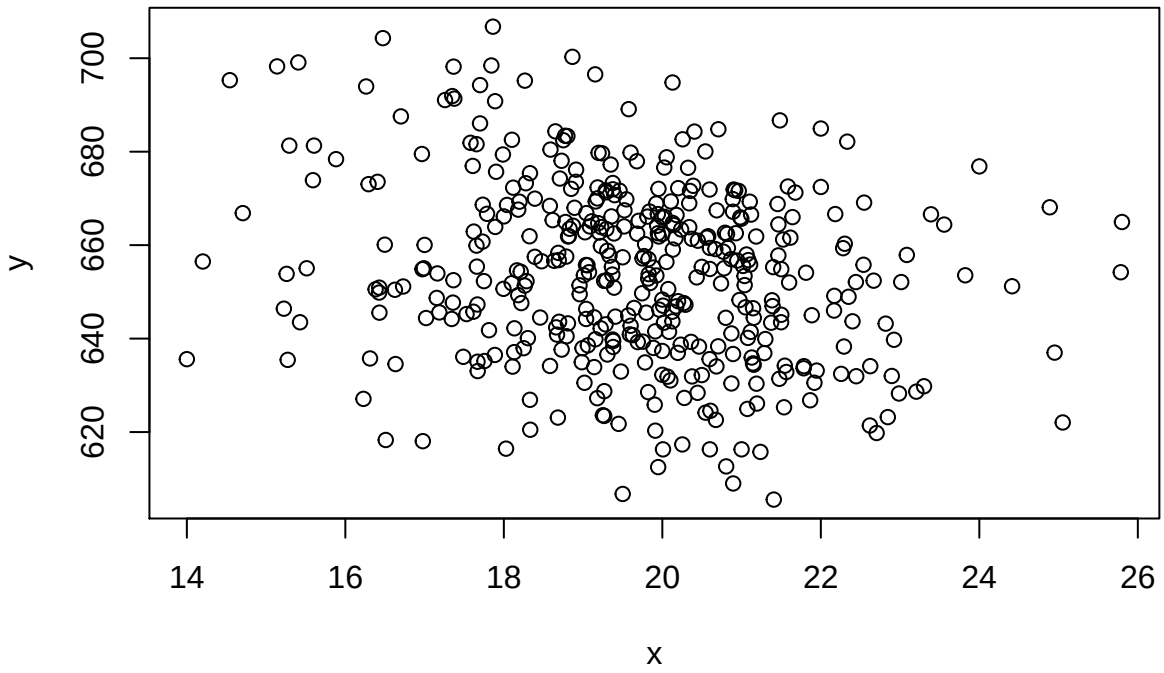

```
plot(x,y,main="Scatterplot for data")
abline(v=hx$mid,col="red") # just to give the picture of what we are doing
```




```
mean.y.given.x = rep(NA,length(hx$mid))
sd.y.given.x = rep(NA,length(hx$mid))
var.y.given.x = rep(NA,length(hx$mid))
for(i in 1:length(hx$mid)){
  if(hx$counts[i]>0){
    mean.y.given.x[i] = mean(y[z==hx$mid[i]])
    sd.y.given.x[i] = sd(y[z==hx$mid[i]])
    var.y.given.x[i] = sd.y.given.x[i]^2
  }}

mean.y.given.x
```

```
##  [1] 672.8833 667.8636 655.7050 663.5526 657.5595 653.3600 653.0059
##  [8] 646.4096 644.7848 650.4286 652.1000 647.0667
```

```
# how to plot both at same time
plot(x, y)
```




```
plot(hx$mid,mean.y.given.x,type="l",xlab="student teacher ratio",ylab="test score",
     col="blue",
     xlim=c(min(x),max(x)),ylim=c(min(y),max(y)))
abline(fit, col="red")

loess.fit = loess(y ~ x)
points(sort(x),predict(loess.fit,data.frame(x=sort(x))),col="green",type="l")
points(x,y,type="p")#,pch="+")
legend('topright',c("linear","binned E(Y|X)","Loess"),
       col=c('red','blue','green'), lty = c(1,1,1))
```