LA-2

Latha Ananya

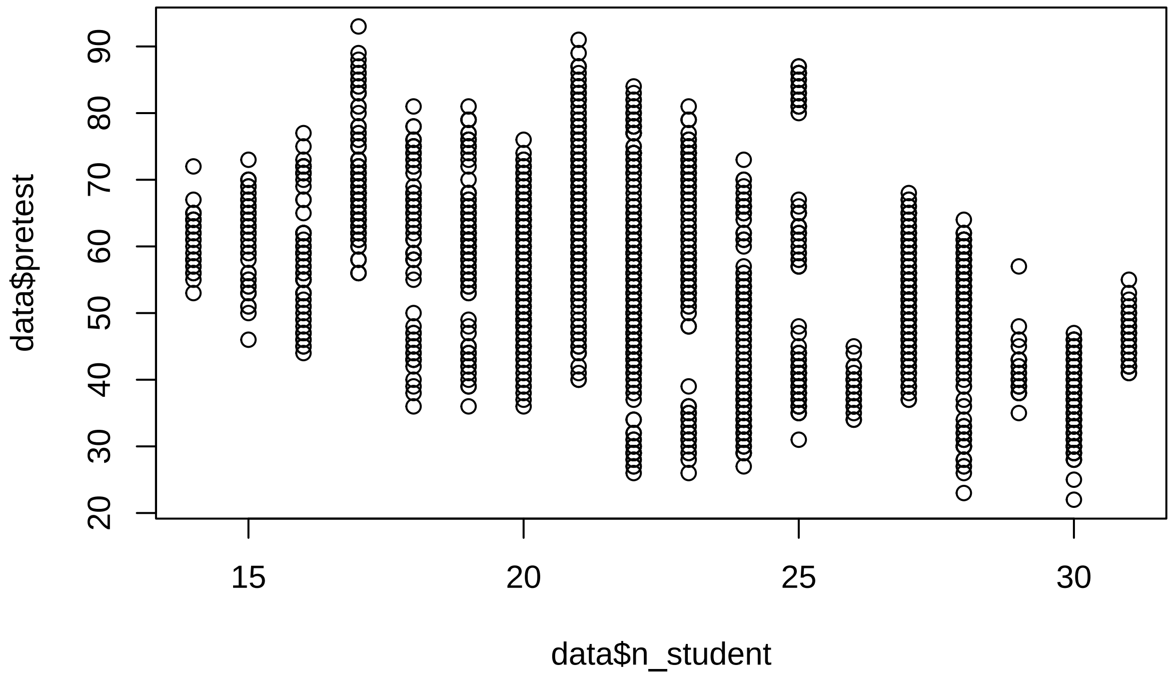
2022-07-02

```
colnames(data)
```

```
##  [1] "school"          "school_setting"  "school_type"     "classroom"
##  [5] "teaching_method" "n_student"       "student_id"      "gender"
##  [9] "lunch"           "pretest"         "posttest"
```

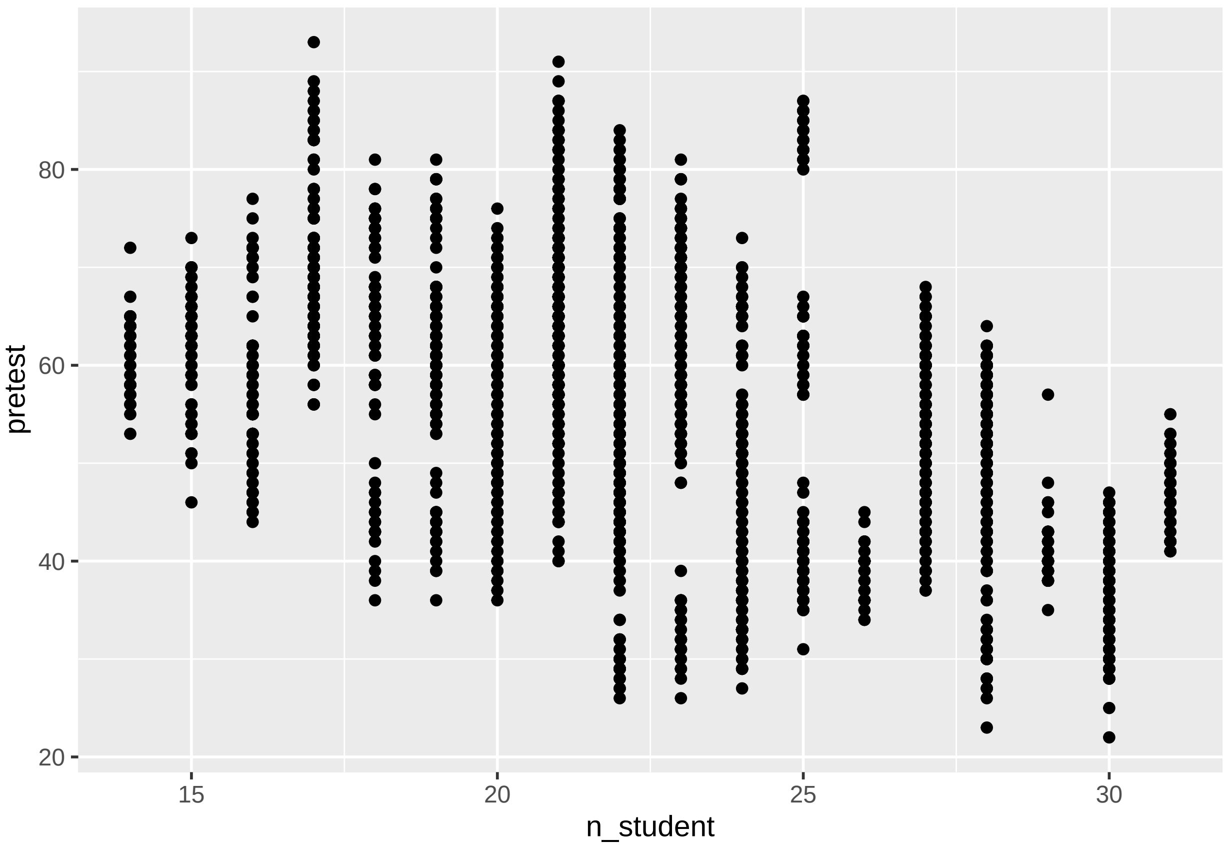
scatter plot plotted for n_student in x axis and pretest in y axis

```
library(ggplot2)
plot(data$n_student,data$pretest)
```



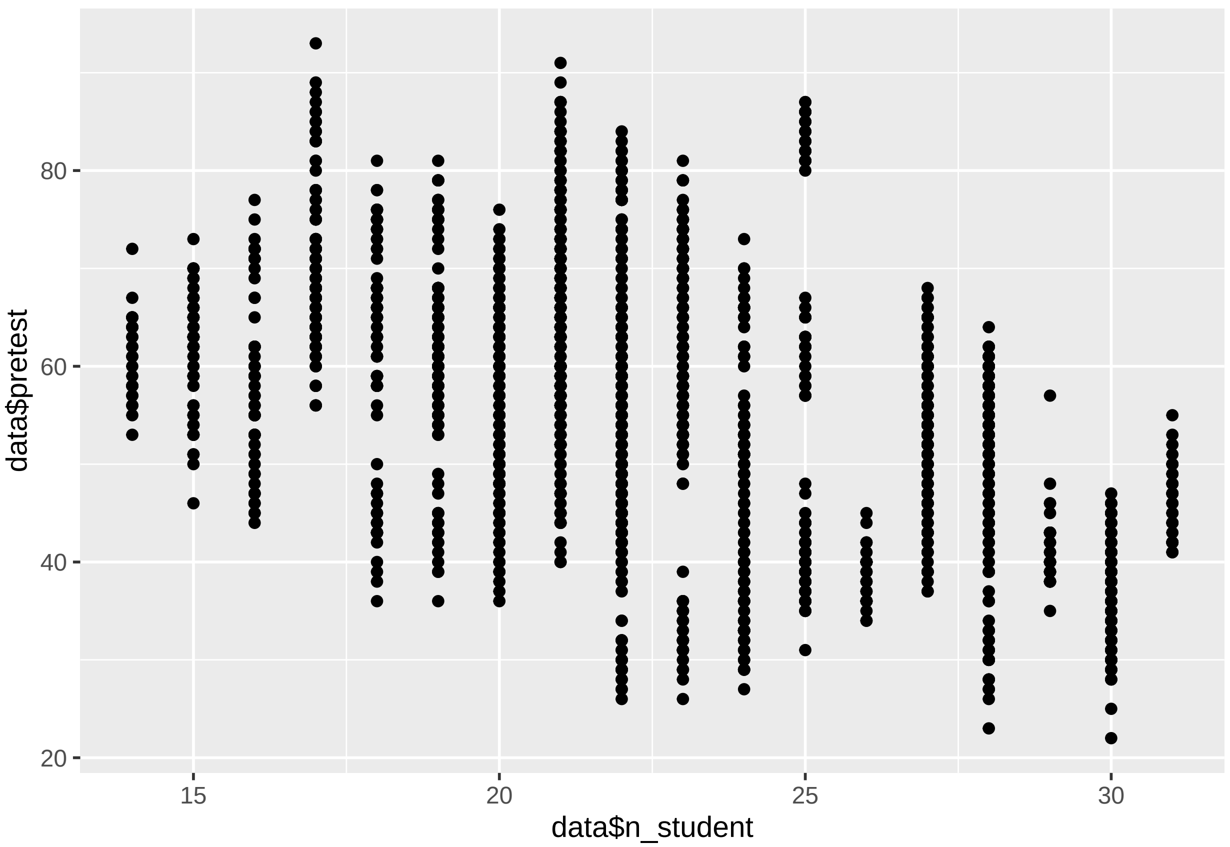
Visualizing the scatter plot using ggplot() function

```
ggplot(data,aes(x=n_student,y=pretest))+geom_point()
```



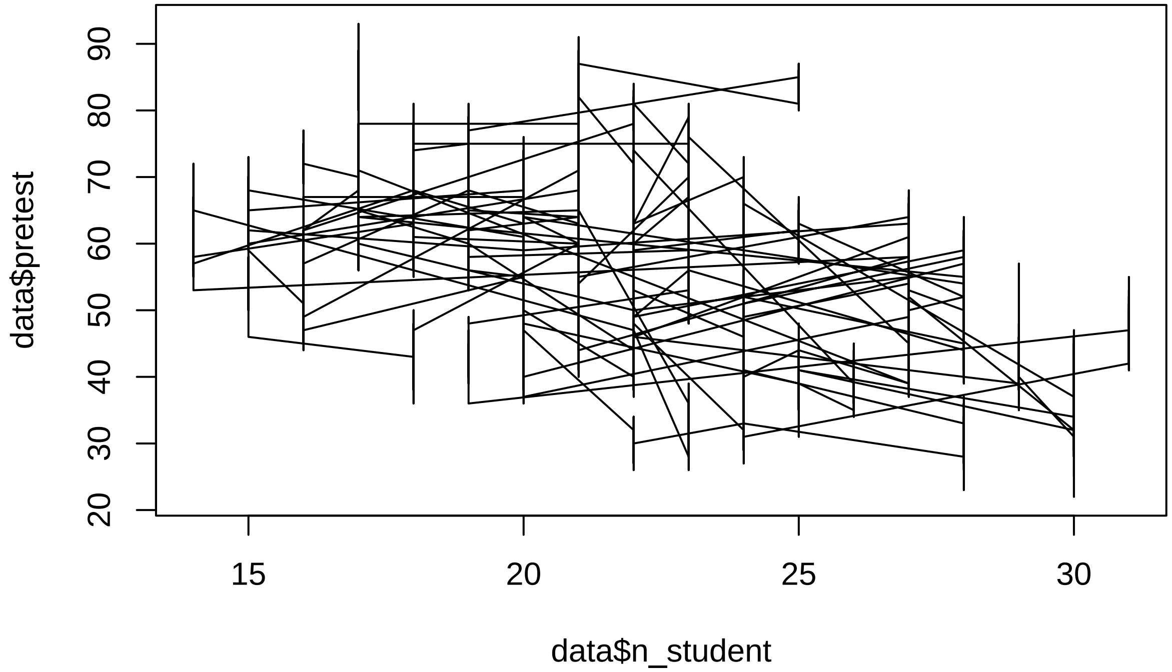
Visualizing the scatter plot using ggplot() function when x and y values are vectors

```
ggplot(NULL,aes(x=data$n_student,y=data$pretest))+geom_point()
```



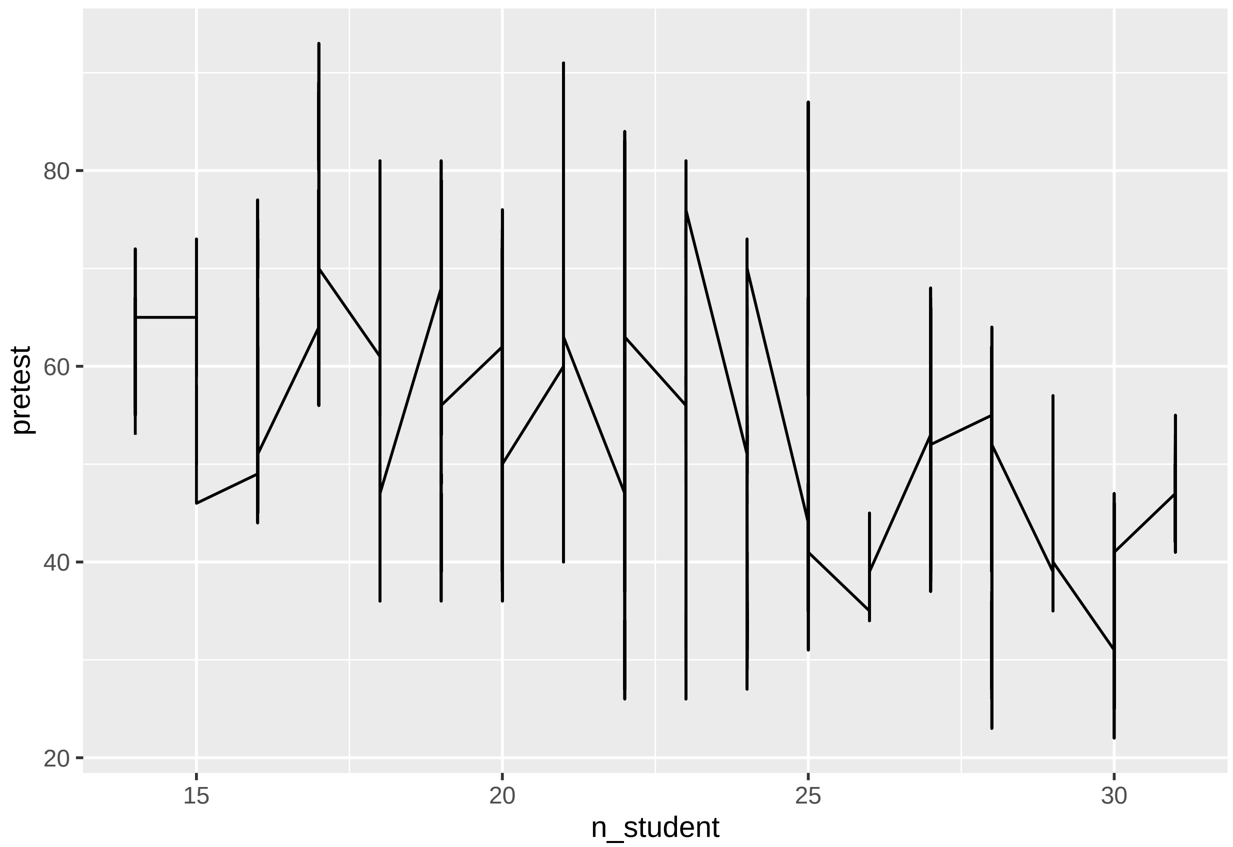
Visualizing the line graph using plot function by passing x and y parameters

```
plot(data$n_student,data$pretest,type="l")
```



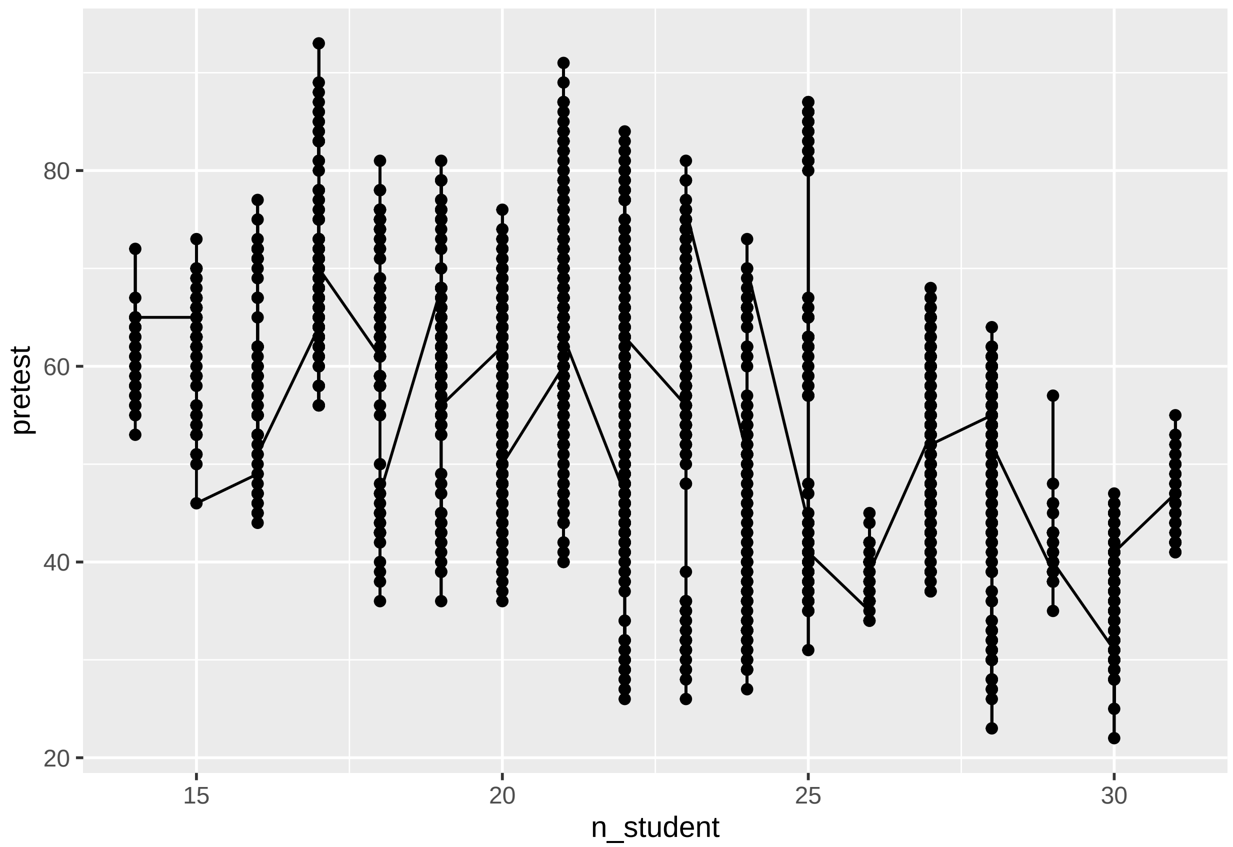
Visualizing the line graph using the ggplot() function

```
ggplot(data,aes(x=n_student,y=pretest)) + geom_line()
```



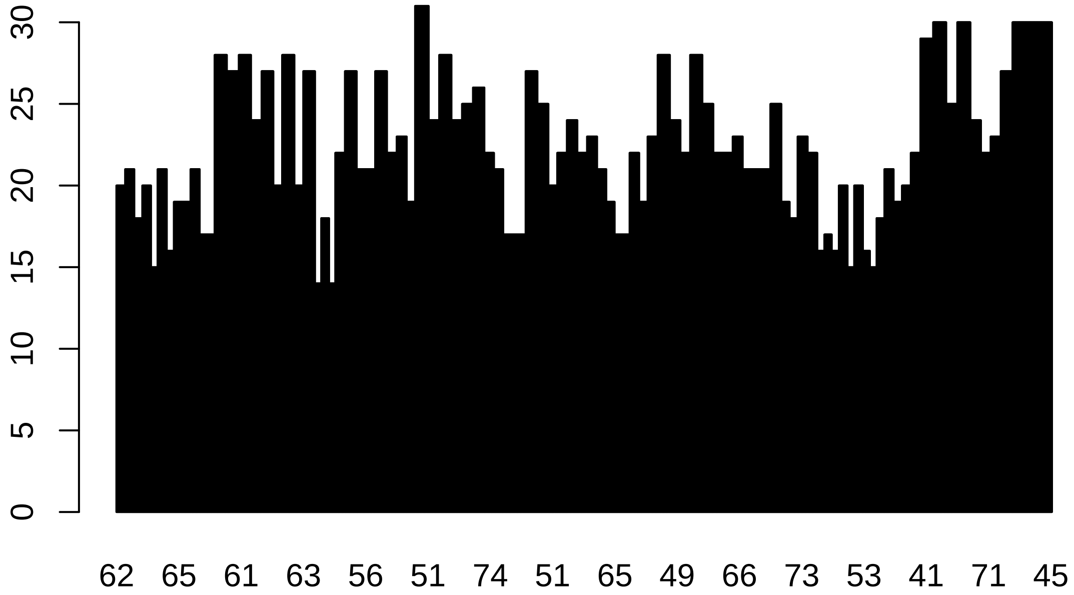
Visualizing the line graph using geom_line() function and plotting the points on top of it

```
ggplot(data,aes(x=n_student,y=pretest)) + geom_line()+geom_point()
```



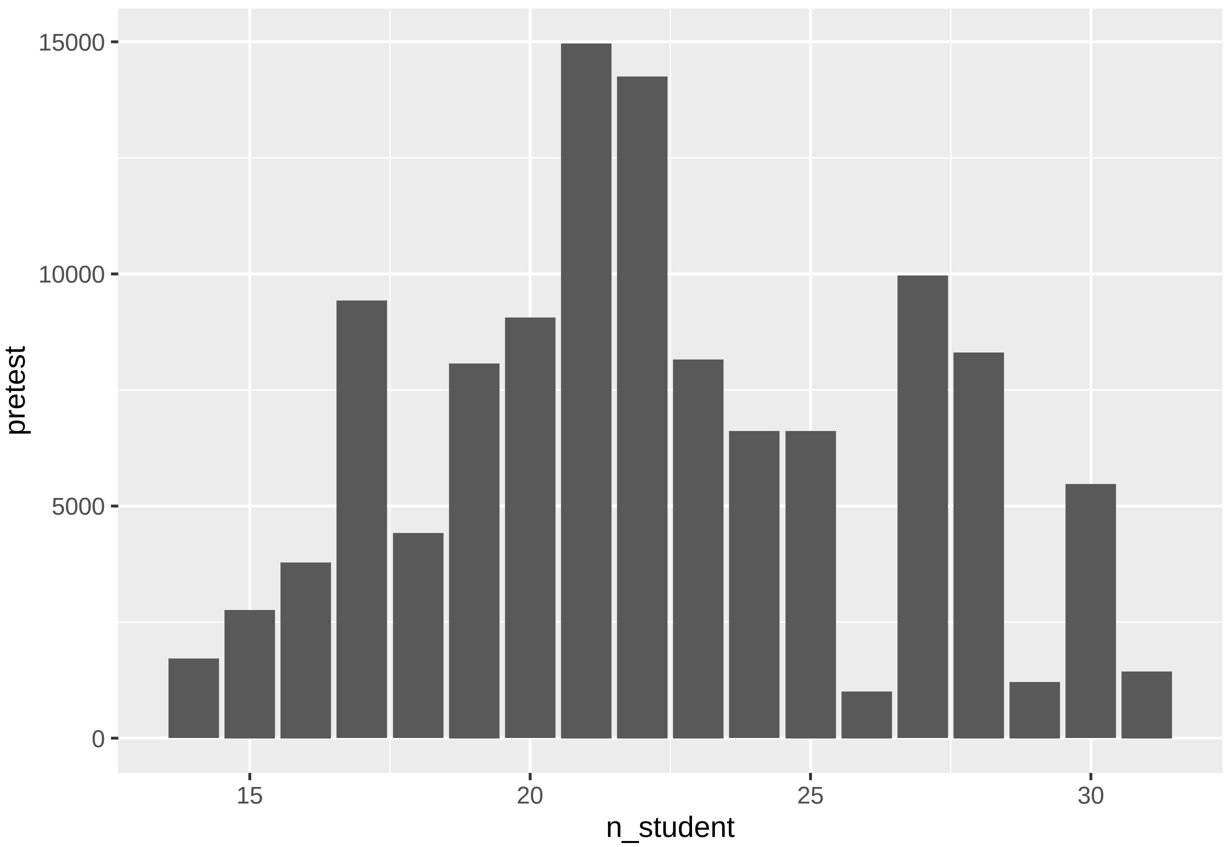
Visualizing the bar graph by plotting x and y axis using barplot function

```
barplot(data$n_student,names.arg=data$pretest)
```



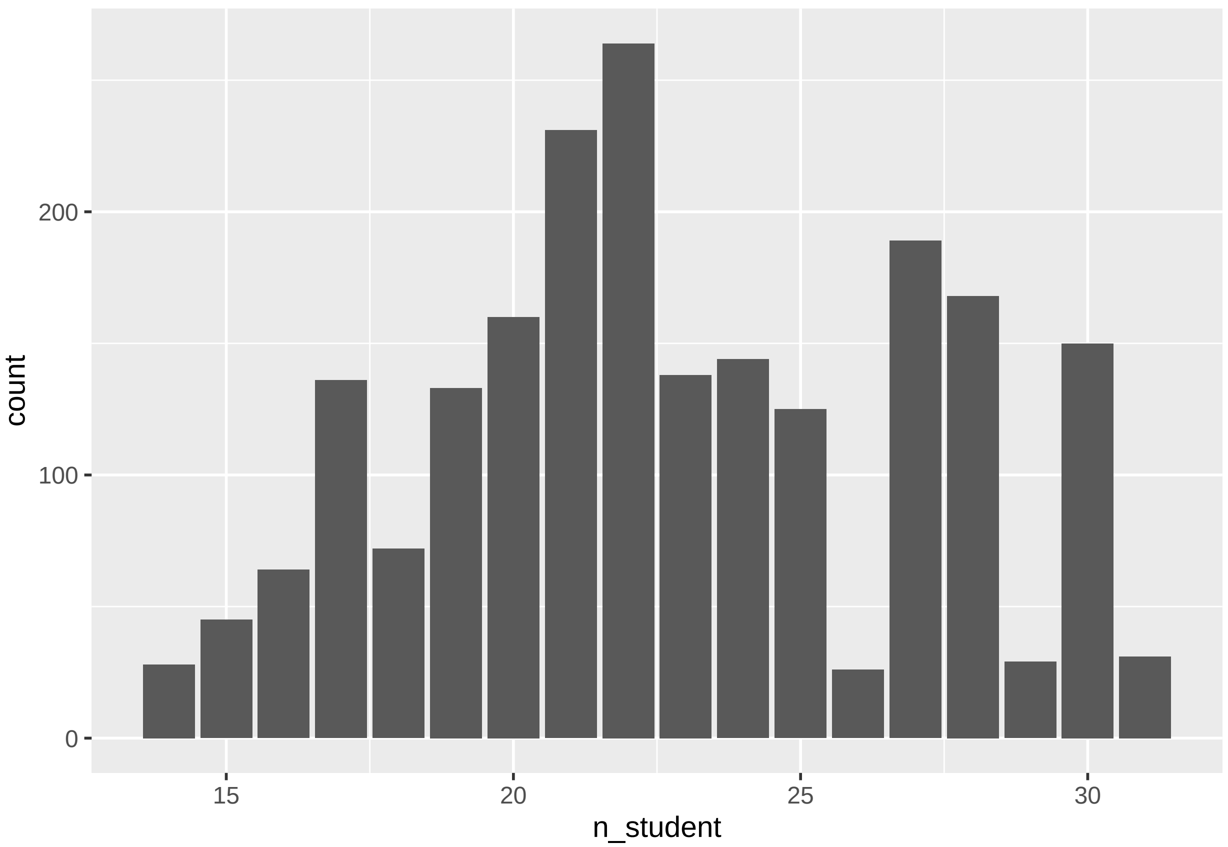
Visualizing the bargraph using ggplot() function

```
ggplot(data,aes(x=n_student,y=pretest))+geom_col()
```



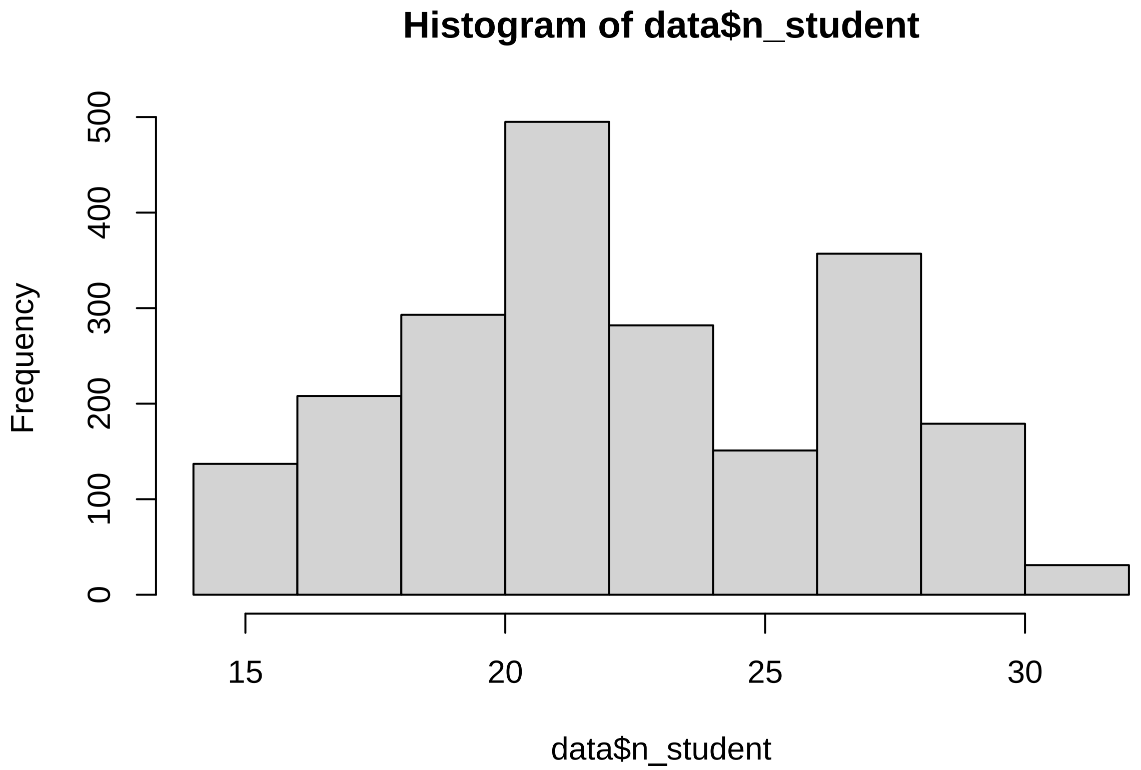
Visualizing the bargraph using ggplot() function using geom_bar() function

```
ggplot(data,aes(x=n_student))+geom_bar()
```



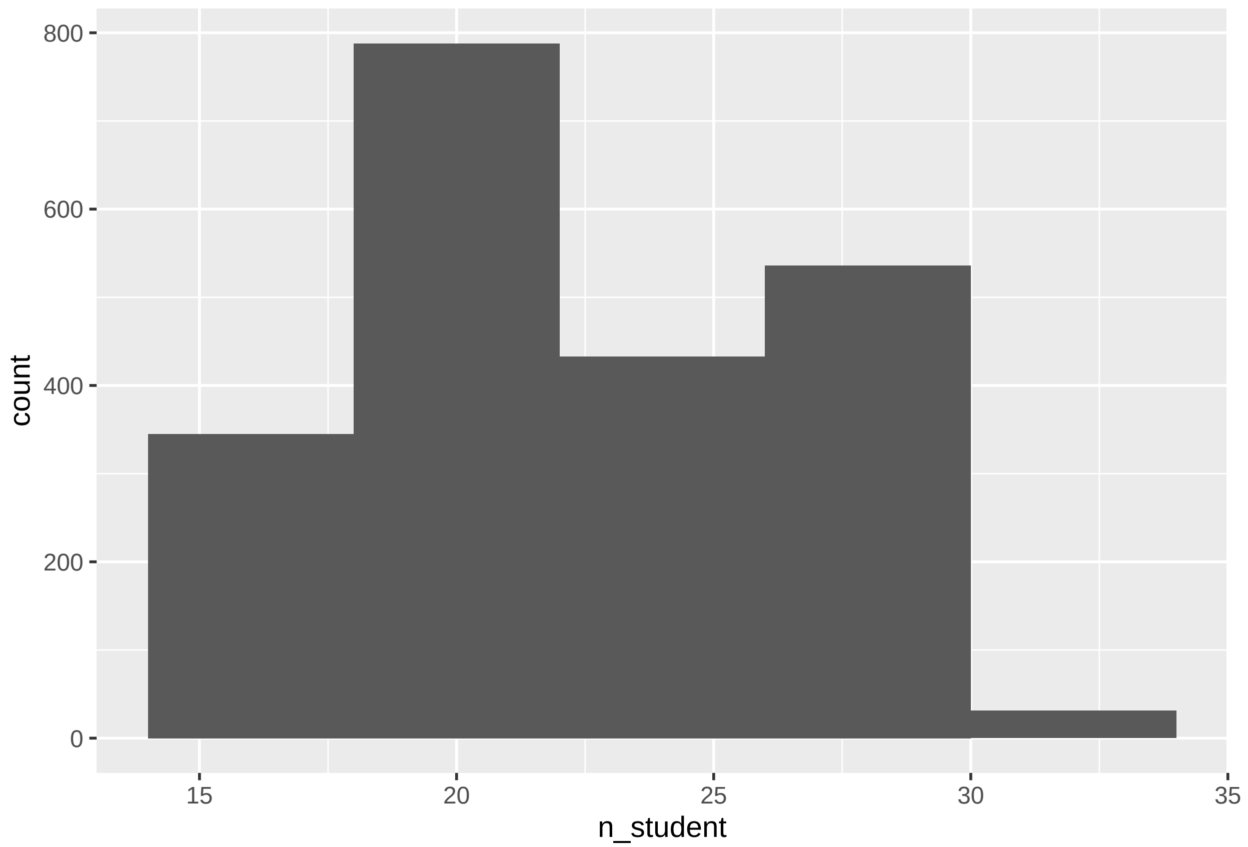
Visualizing the histograms using the hist() function

```
hist(data$n_student,breaks=10)
```



Visualizing the histograms using the ggplot() function

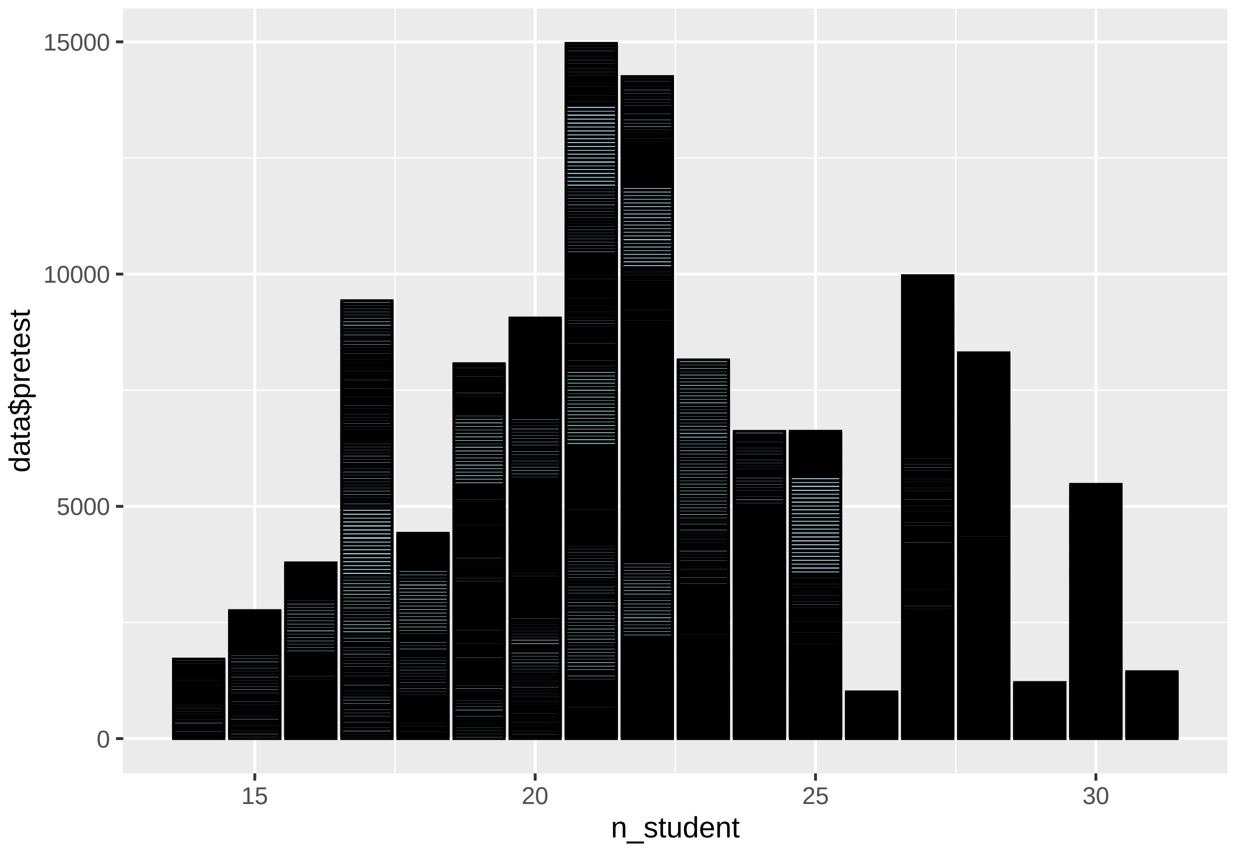
```
ggplot(data,aes(x=n_student)) + geom_histogram(binwidth = 4)
```



Visualizing the bargraph using ggplot() function and it fills the light blue colour to the bars and the border lines of the bar will be black

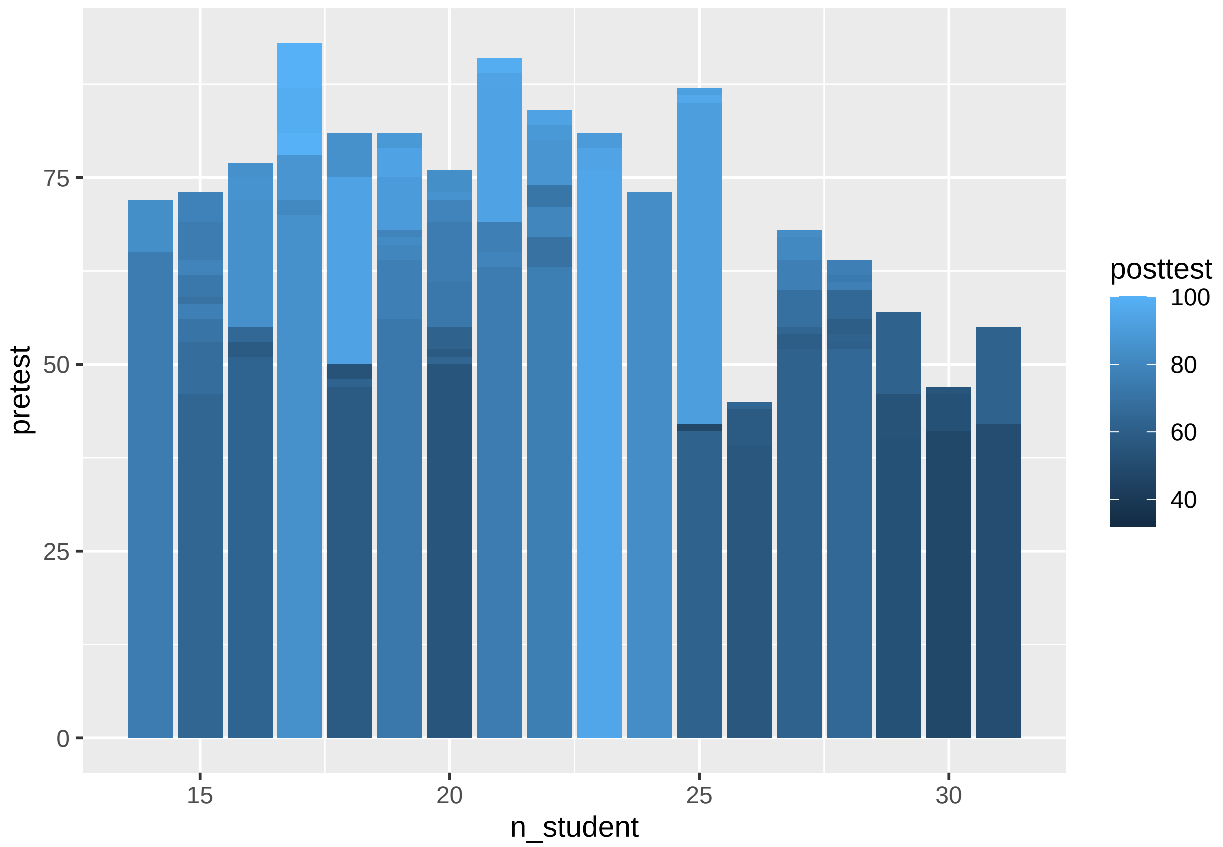
```
ggplot(data,aes(x=n_student,y=data$pretest)) + geom_col(fill="lightblue",colour="black")
```

```
## Warning: Use of ‘data$pretest‘ is discouraged. Use ‘pretest‘ instead.
```



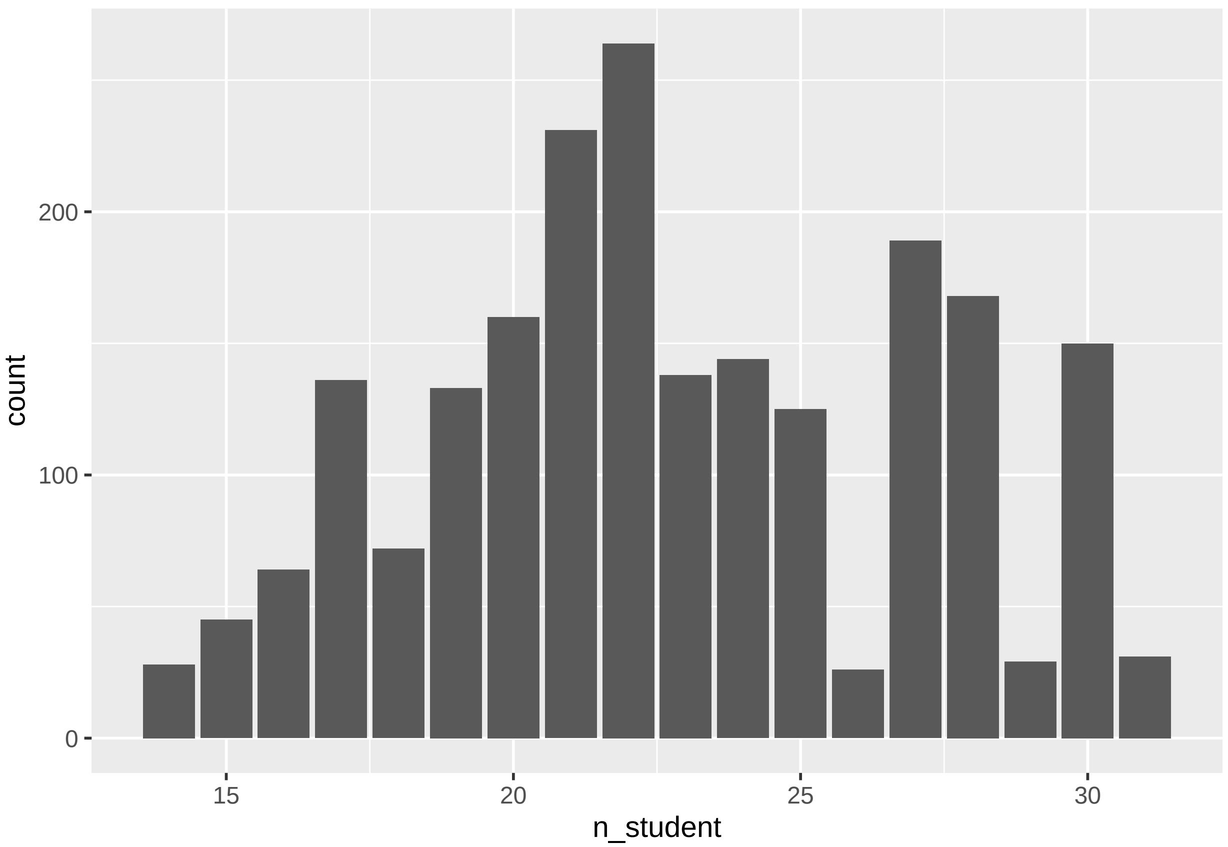
Visualizing the bargraph and mapping the posttest coloumn to the fill colour

```
ggplot(data,aes(x=n_student,y=pretest,fill=posttest))+geom_col(position="dodge")
```



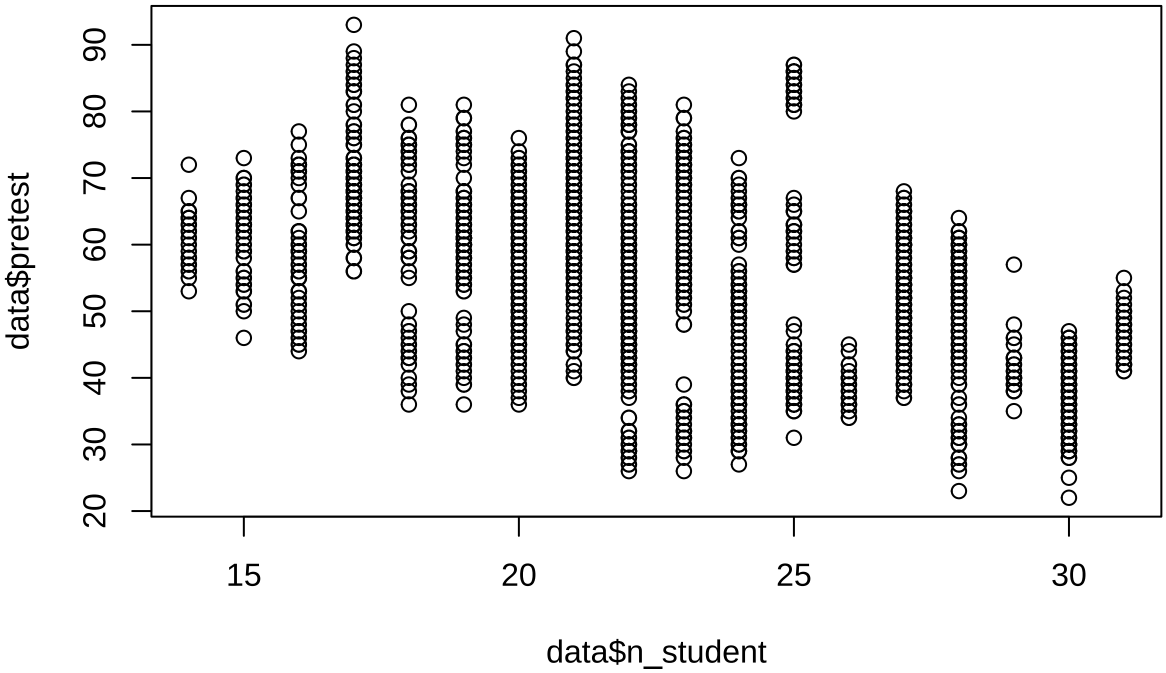
Bar of points

```
ggplot(data,aes(x=n_student))+geom_bar()
```



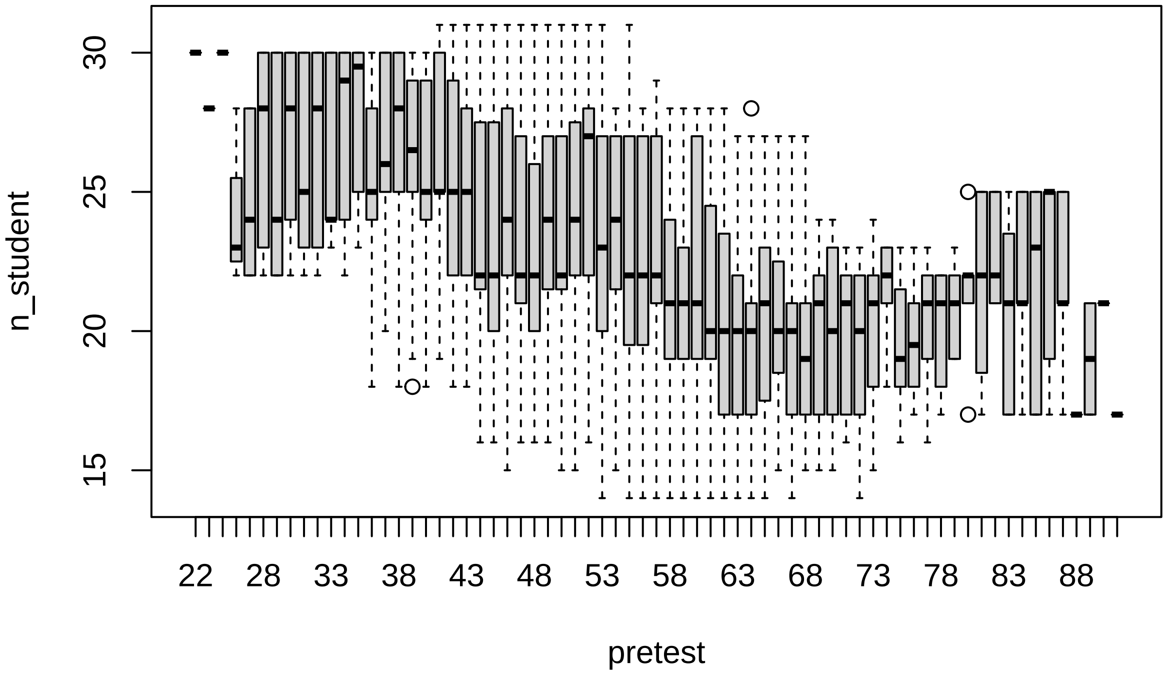
Creating a boxplot using boxplot() function

```
plot(data$n_student, data$pretest)
```



Put interaction of two variables on x-axis

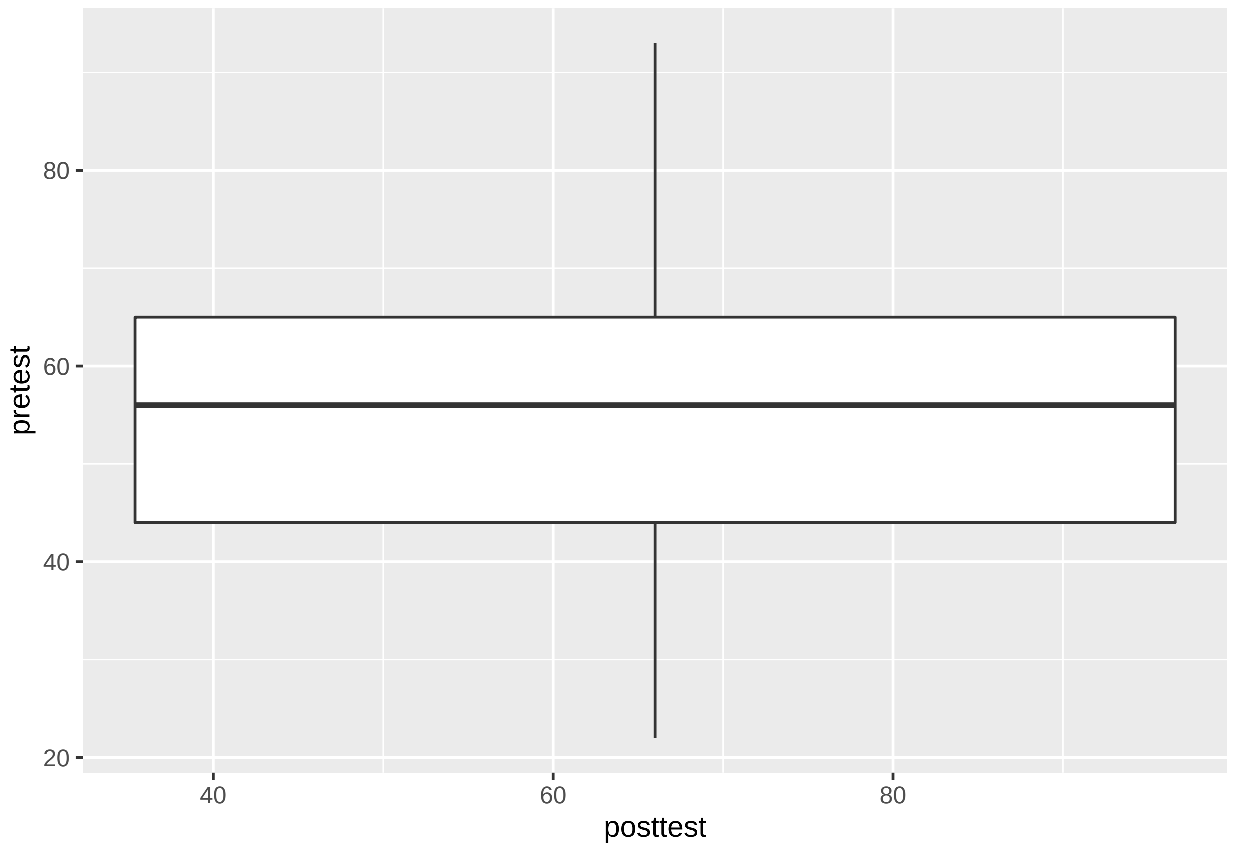
```
boxplot(n_student ~  + pretest, data = data)
```



Visualization of boxplot using the ggplot() function

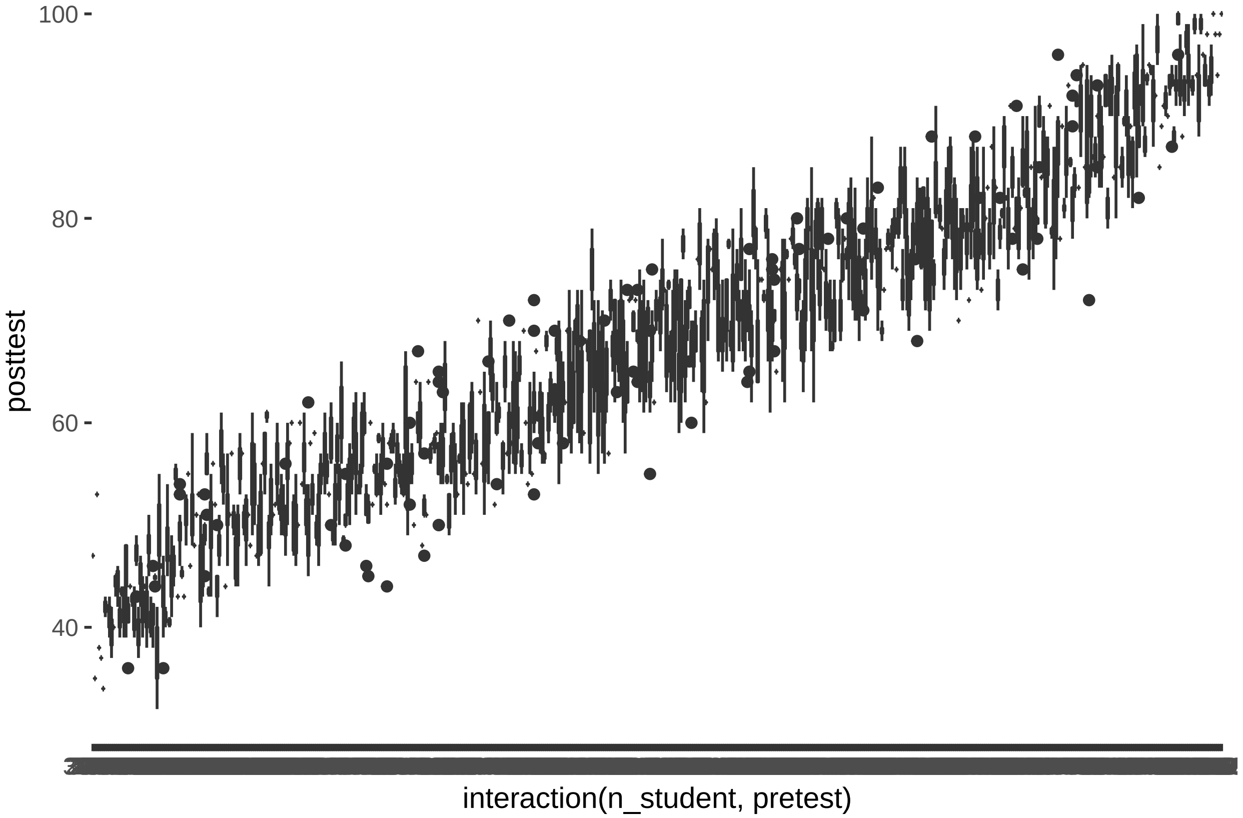
```
ggplot(data, aes(x = posttest, y = pretest)) + geom_boxplot()
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



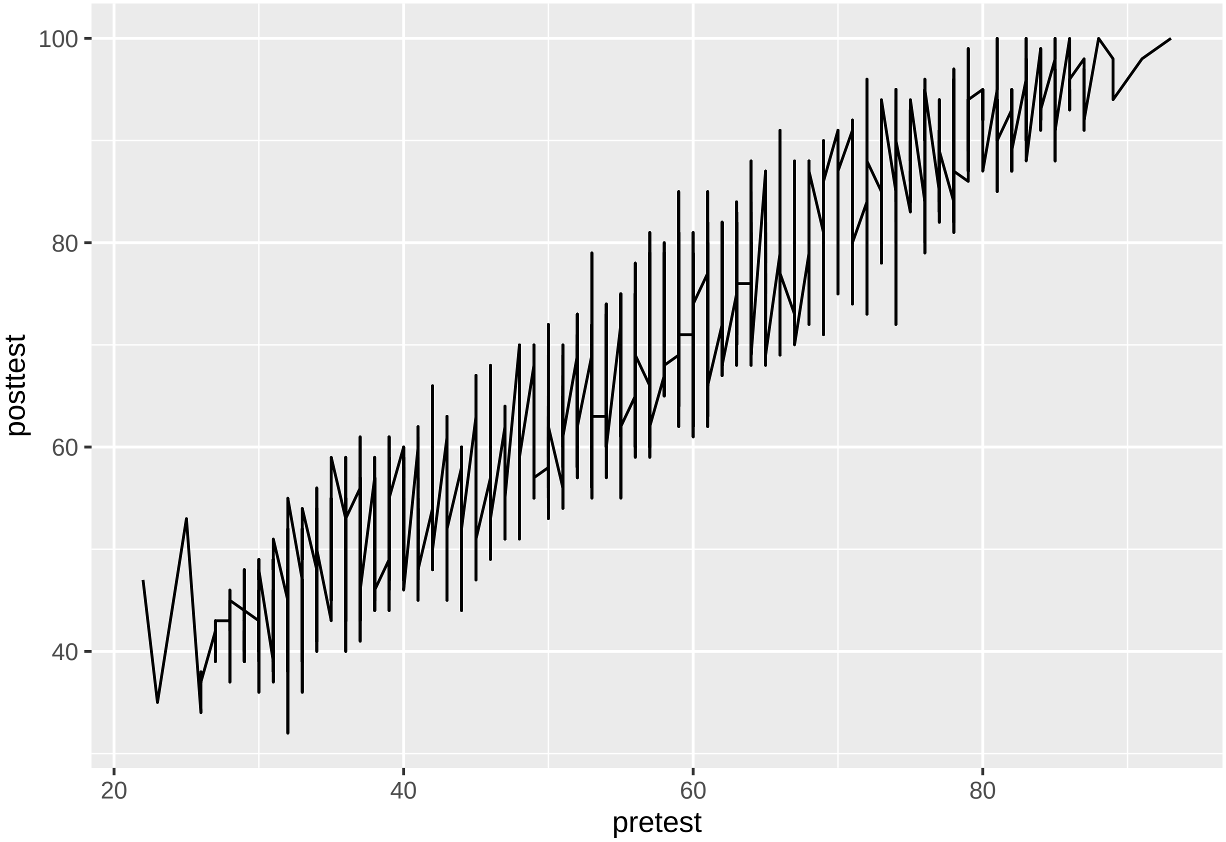
Make box plots for multiple variables, by combining the variables with interaction(),

```
ggplot(data, aes(x = interaction(n_student, pretest), y = posttest)) + geom_boxplot()
```



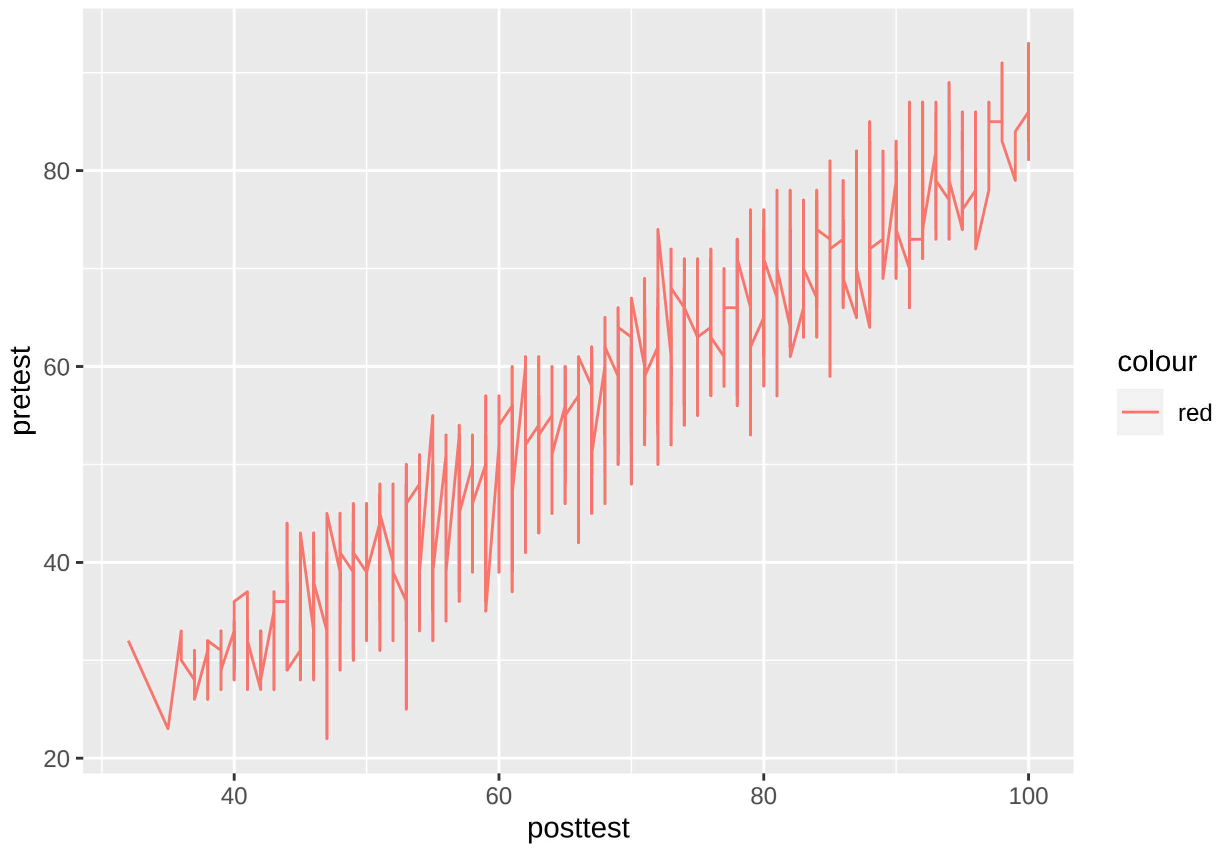
When the x variable is a factor, you must also use aes(group=1) to ensure that ggplot knows that the data points belong together and should be connected with a line

```
ggplot(data, aes(x = pretest, y = posttest, group = 1)) + geom_line()
```



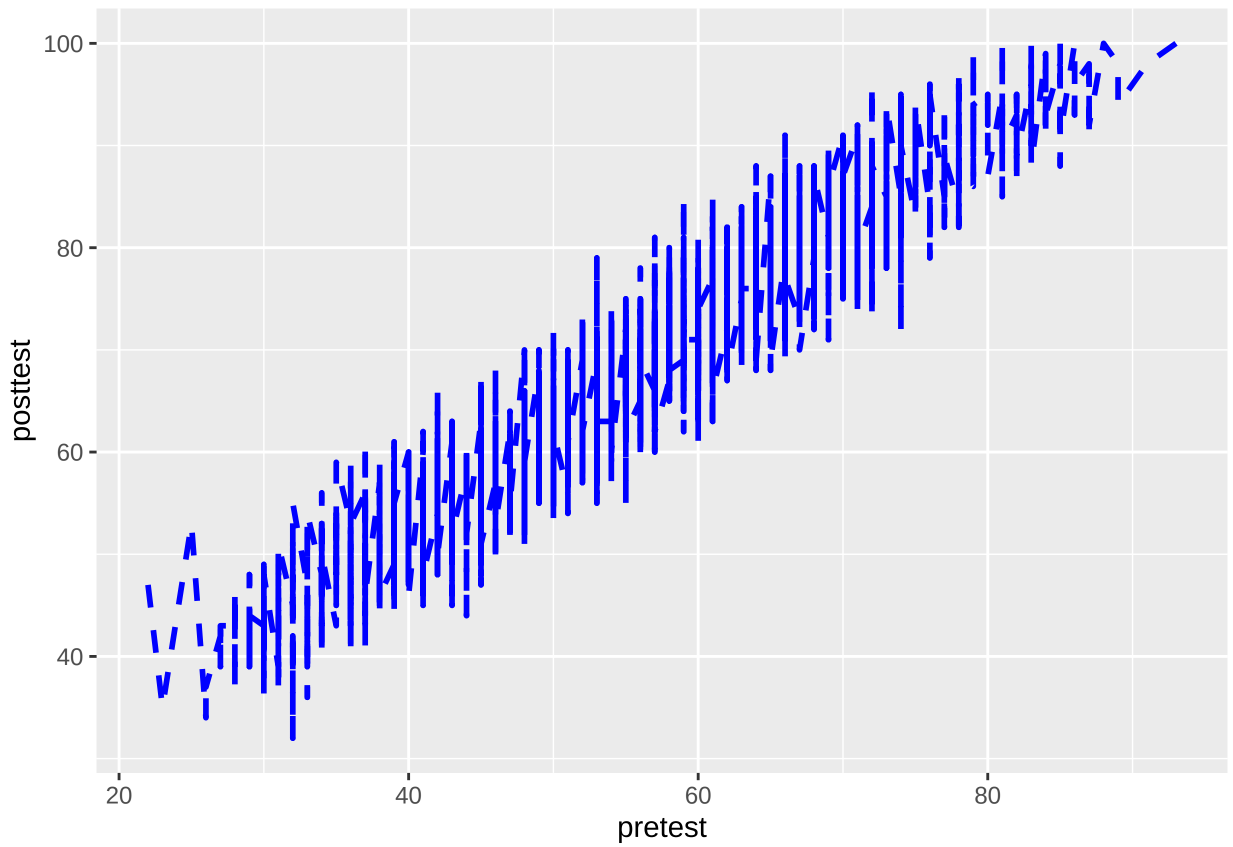
Variables mapped to the x- and y-axes, map another (discrete) vari- able to colour or linetype

```
ggplot(data, aes(x = posttest, y = pretest, colour = "red")) + geom_line()
```



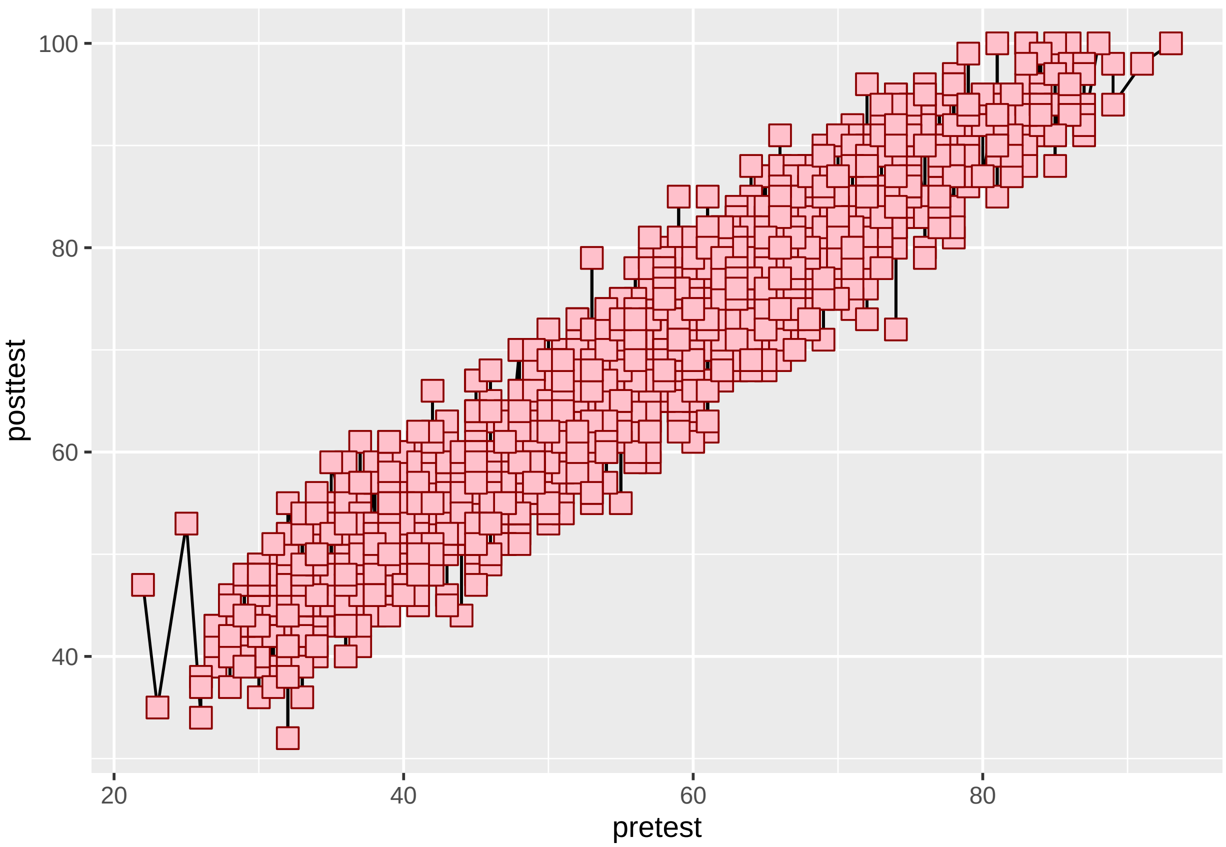
Changing the appearnce of the lines

```
ggplot(data, aes(x = pretest, y = posttest)) +
 geom_line(linetype = "dashed", size = 1, colour = "blue")
```



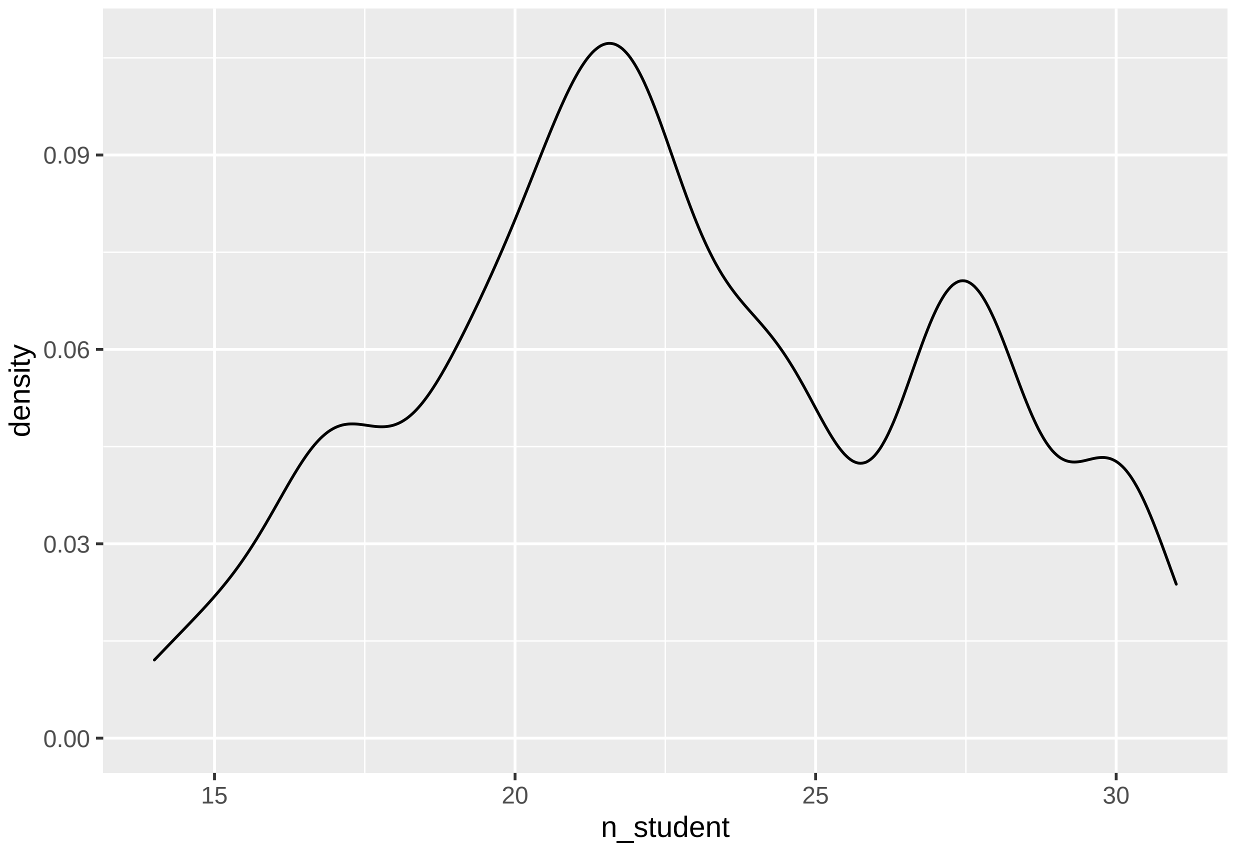
In geom_point(), set the size, shape, colour, and/or fill outside of aes() (the result is shown

```
ggplot(data, aes(x = pretest, y = posttest)) + geom_line() +
 geom_point(size = 4, shape = 22, colour = "darkred", fill = "pink")
```



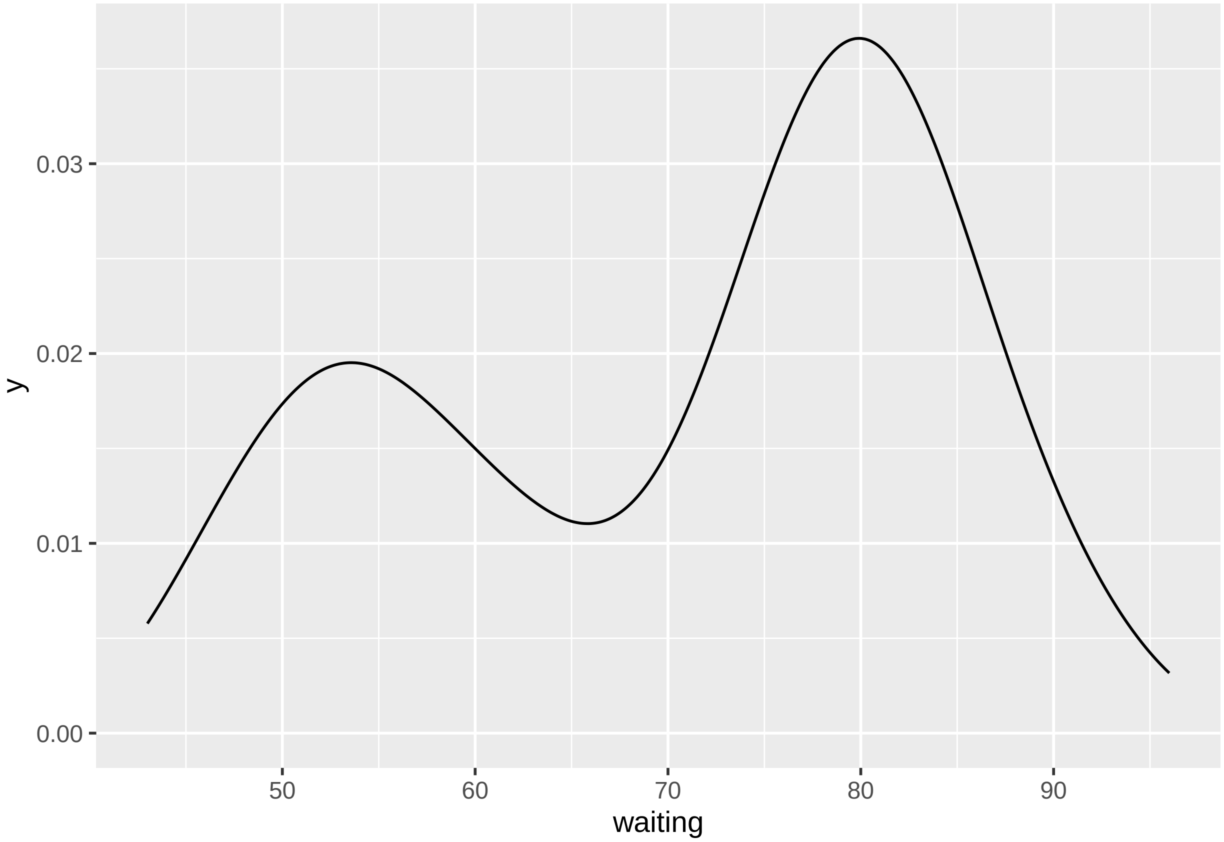
Visualizing the density curve using geom_density() fucntion and map a continuous variable n_student to x

```
ggplot(data, aes(x = n_student)) +
 geom_density()
```



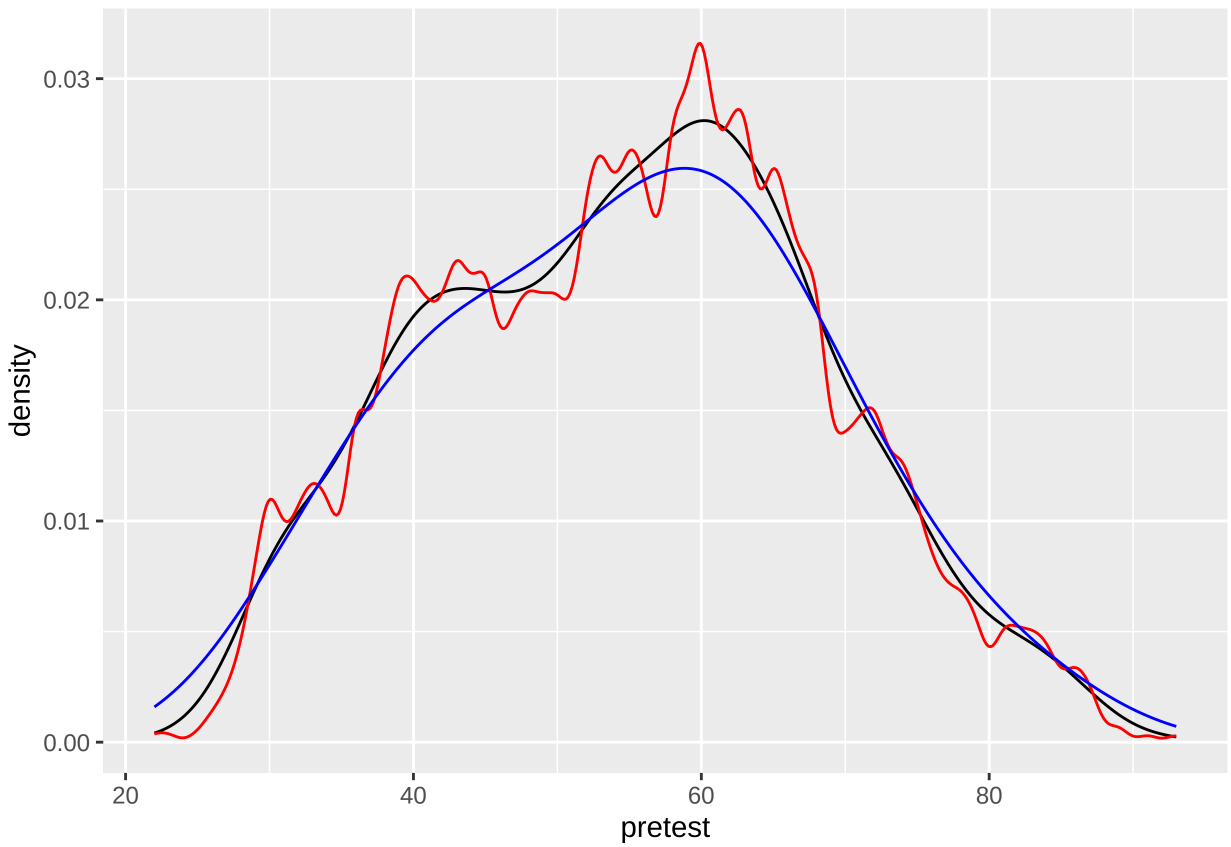
It removes the lines along the bottom and side and using expand_limits() increases the y range to include the value 0

```
ggplot(faithful, aes(x = waiting)) + geom_line(stat = "density") +
 expand_limits(y = 0)
```



Visualizing the density curve using the geom_line function and using a adjust parameter which is used to get the extent to which the density curve should be smoother

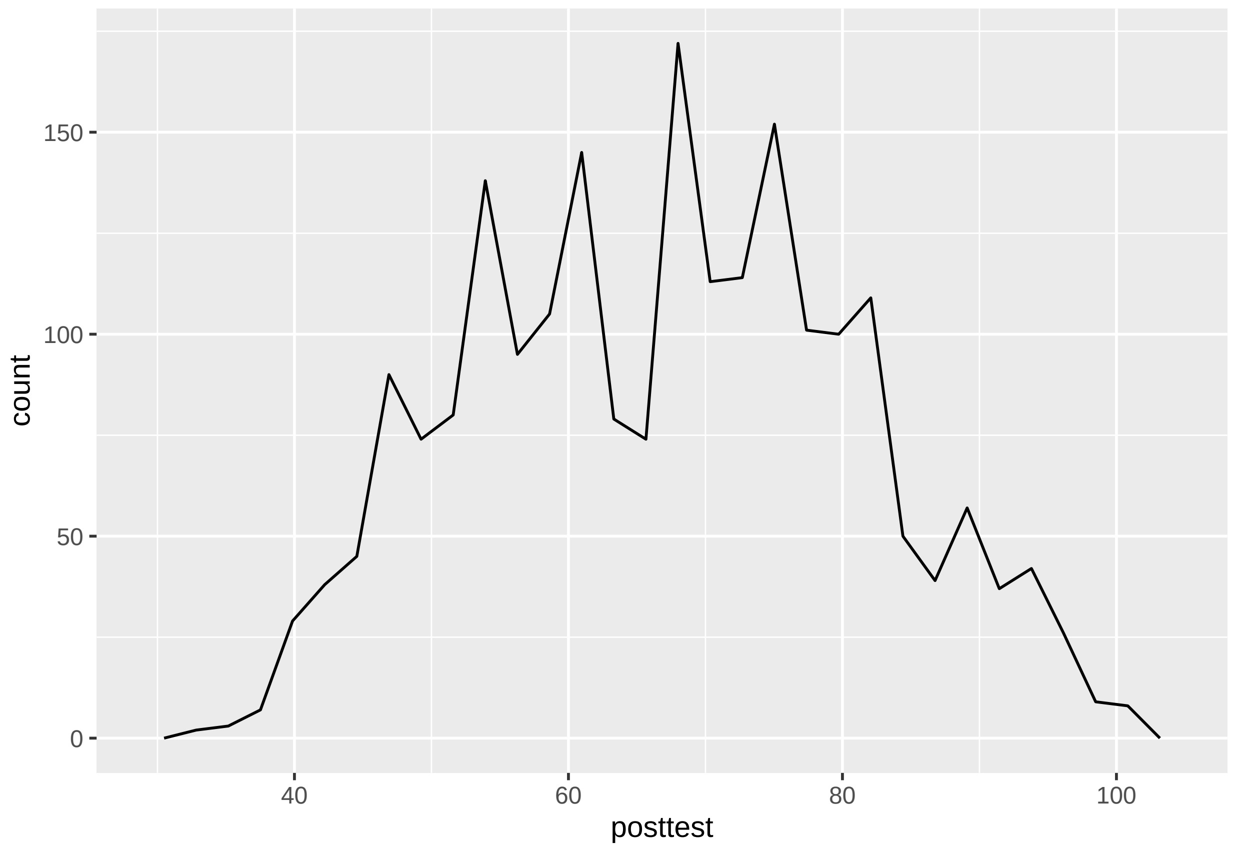
```
ggplot(data, aes(x = pretest)) +
 geom_line(stat = "density") +
 geom_line(stat = "density", adjust = .25, colour = "red") +
 geom_line(stat = "density", adjust = 2, colour = "blue")
```



Visualizing the frequency polygon

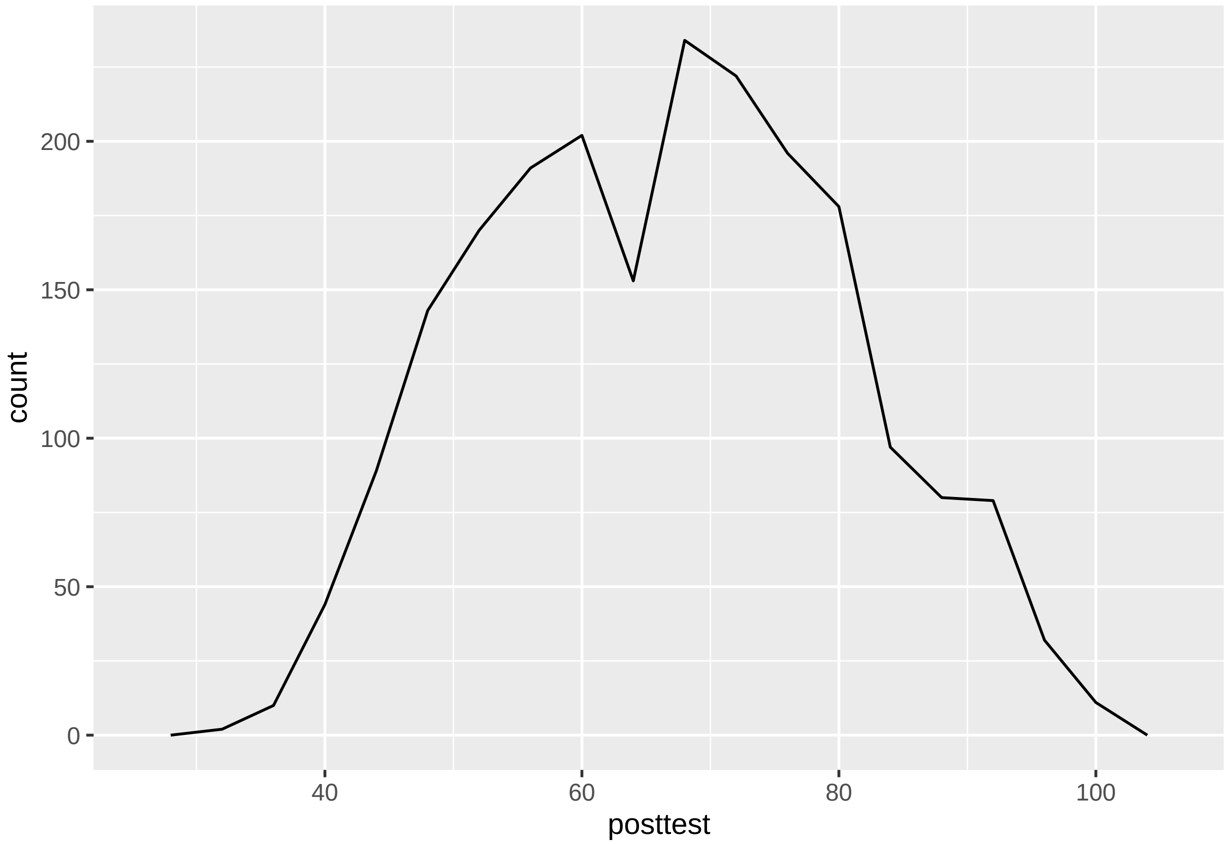
```
ggplot(data, aes(x=posttest)) +  geom_freqpoly()
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



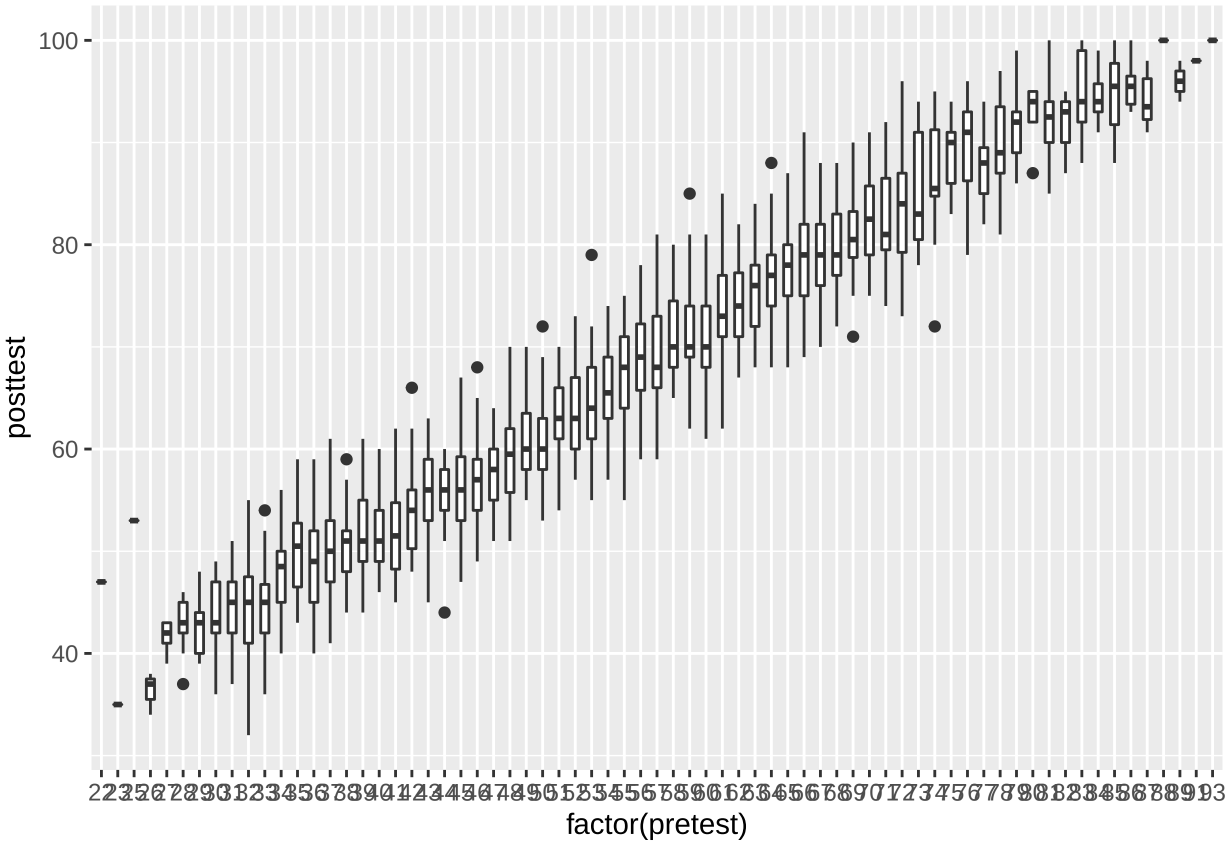
Visualizing the frequency polygon along with the bandwidth parameter

```
ggplot(data, aes(x = posttest)) +  geom_freqpoly(binwidth = 4)
```



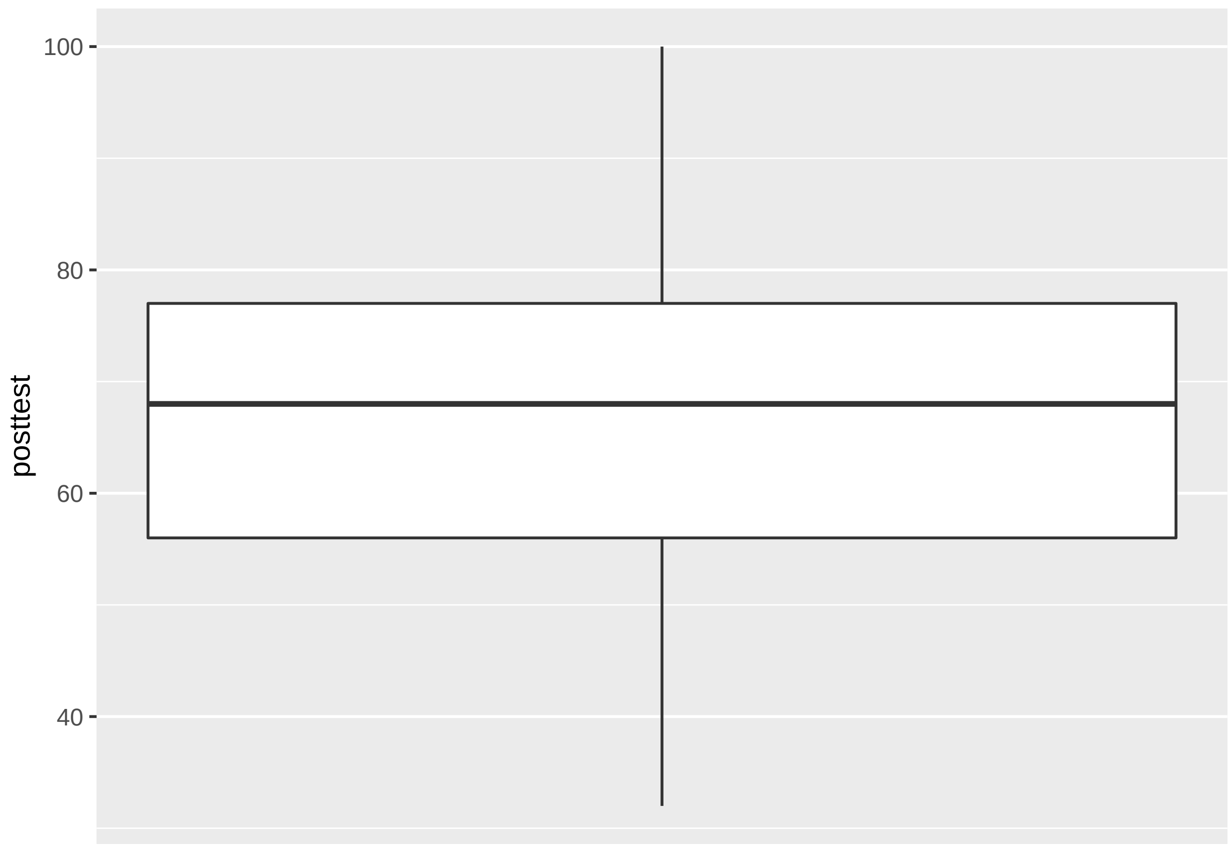
Visualization of the boxplot by changing the width of the bars

```
ggplot(data, aes(x = factor(pretest), y = posttest)) + geom_boxplot(width = .5)
```



Visualization of the a box plot of just a single group, we have to provide some arbitrary value for x.In this case,we'll set it to 1 and remove the x-axis tick markers and label

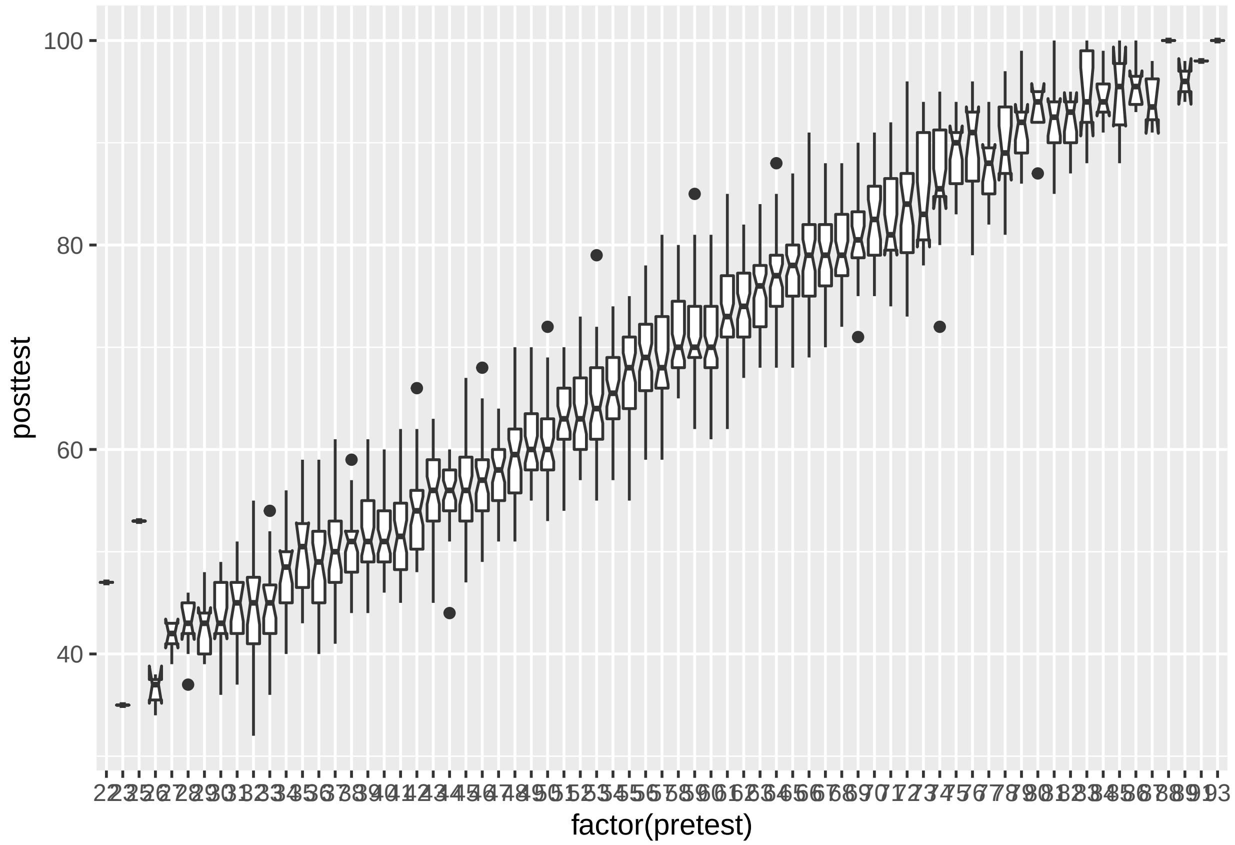
```
ggplot(data, aes(x = 1, y = posttest)) +
 geom_boxplot() +
 scale_x_continuous(breaks = NULL) +
 theme(axis.title.x = element_blank())
```



Adding the notches to the box plot using geom_boxplot() and set notch = TRUE

```
ggplot(data, aes(x = factor(pretest), y = posttest)) +
 geom_boxplot(notch = TRUE)
```

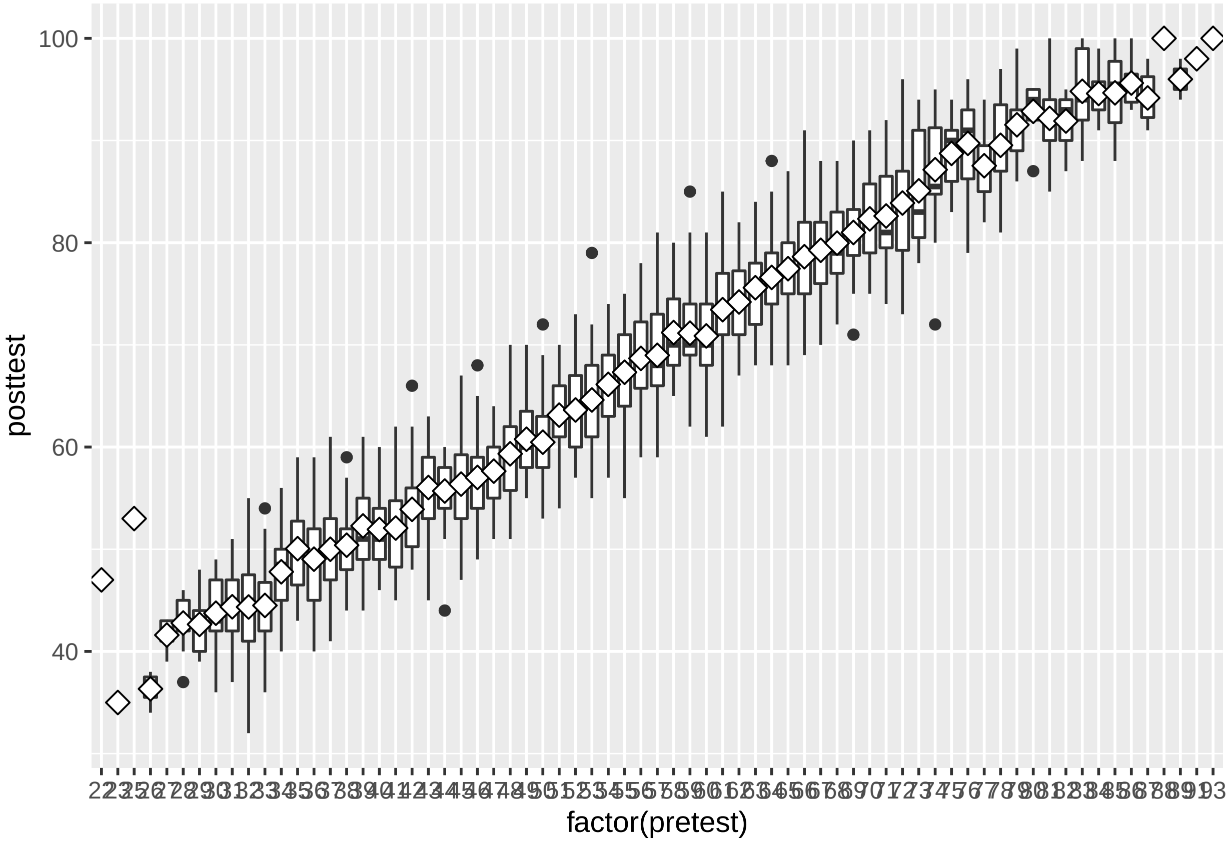
```
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
## notch went outside hinges. Try setting notch=FALSE.
```



Visualizing the box plot by adding the stat_summary function which includes the function of mean which leads to adding of the means to the boxplot

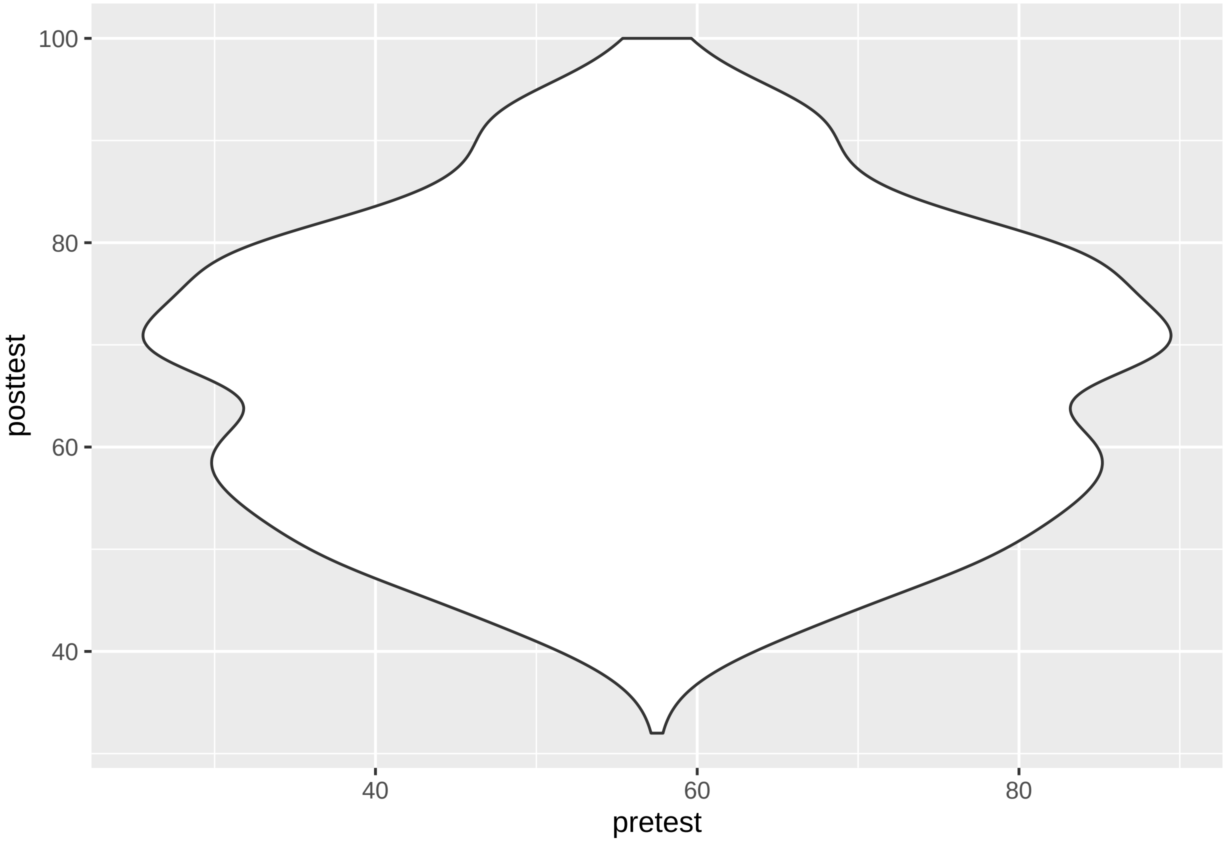
```
ggplot(data, aes(x = factor(pretest), y = posttest)) +
 geom_boxplot() +
 stat_summary(fun.y = "mean", geom = "point", shape = 23, size = 3,
 fill = "white")
```

```
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



Visualization of the violin plot

```
ggplot(data, aes(x = pretest, y = posttest)) + geom_violin()
```

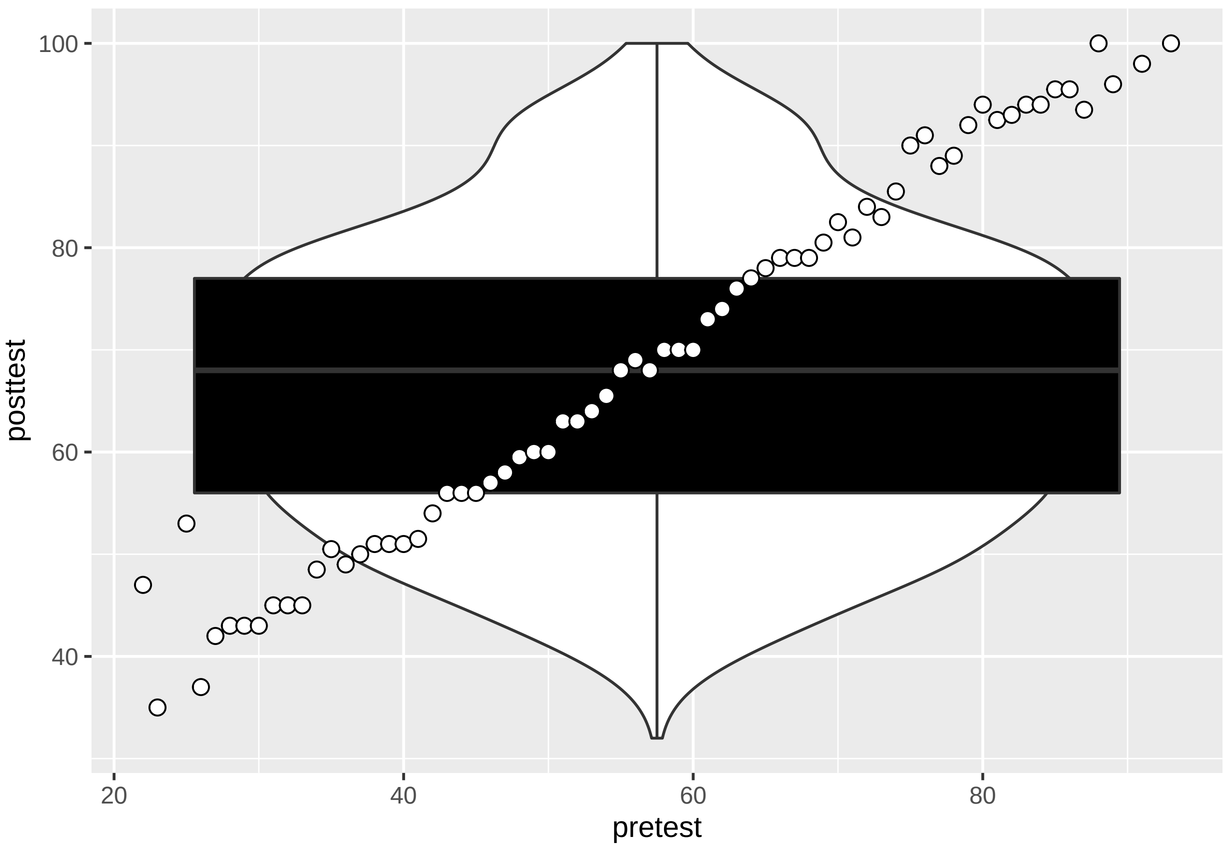


Visualization of the violin plot with the boxplpot overlaid on it

```
ggplot(data,aes(x=pretest,y=posttest))+ geom_violin() +
 geom_boxplot(width = .1, fill = "black", outlier.colour = NA) +
 stat_summary(fun.y = median, geom = "point", fill = "white", shape = 21,
 size = 2.5)
```

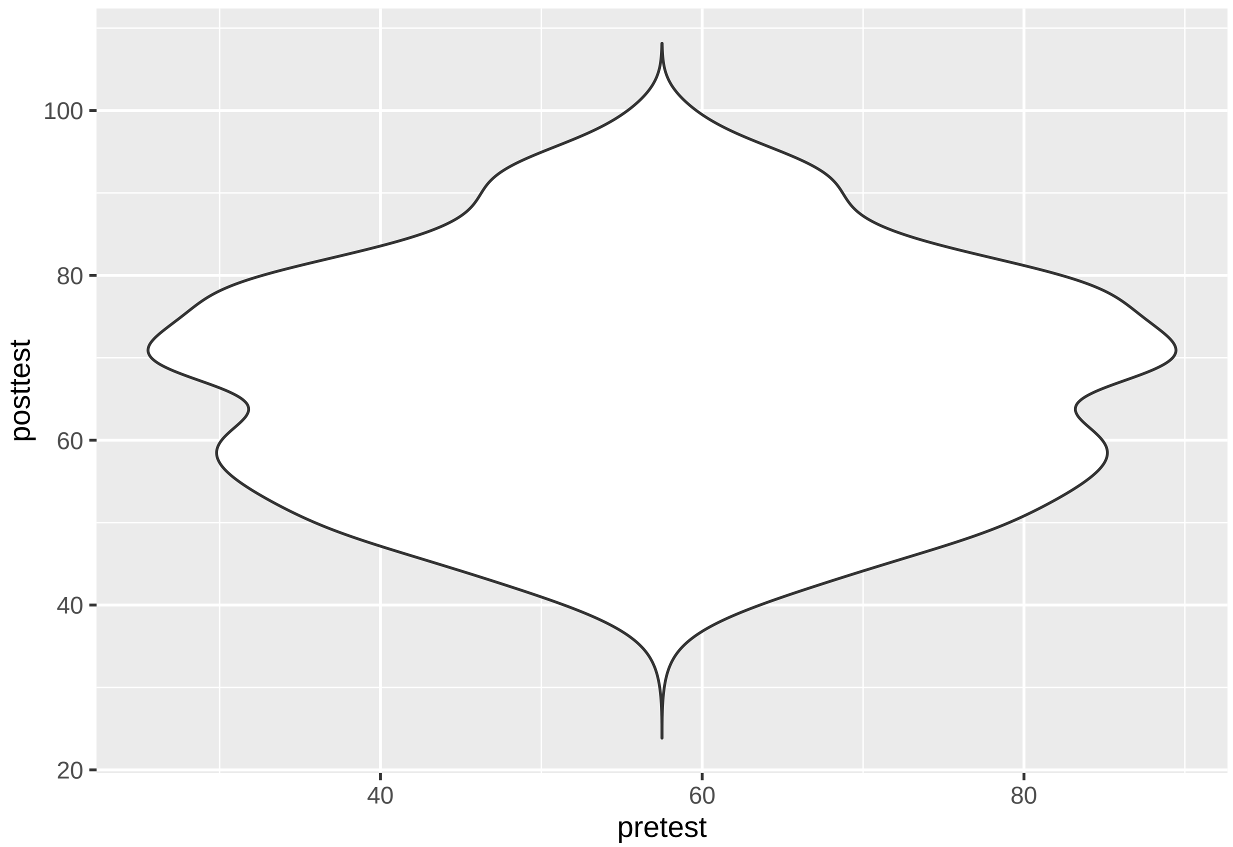
```
## Warning: ‘fun.y‘ is deprecated. Use ‘fun‘ instead.
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



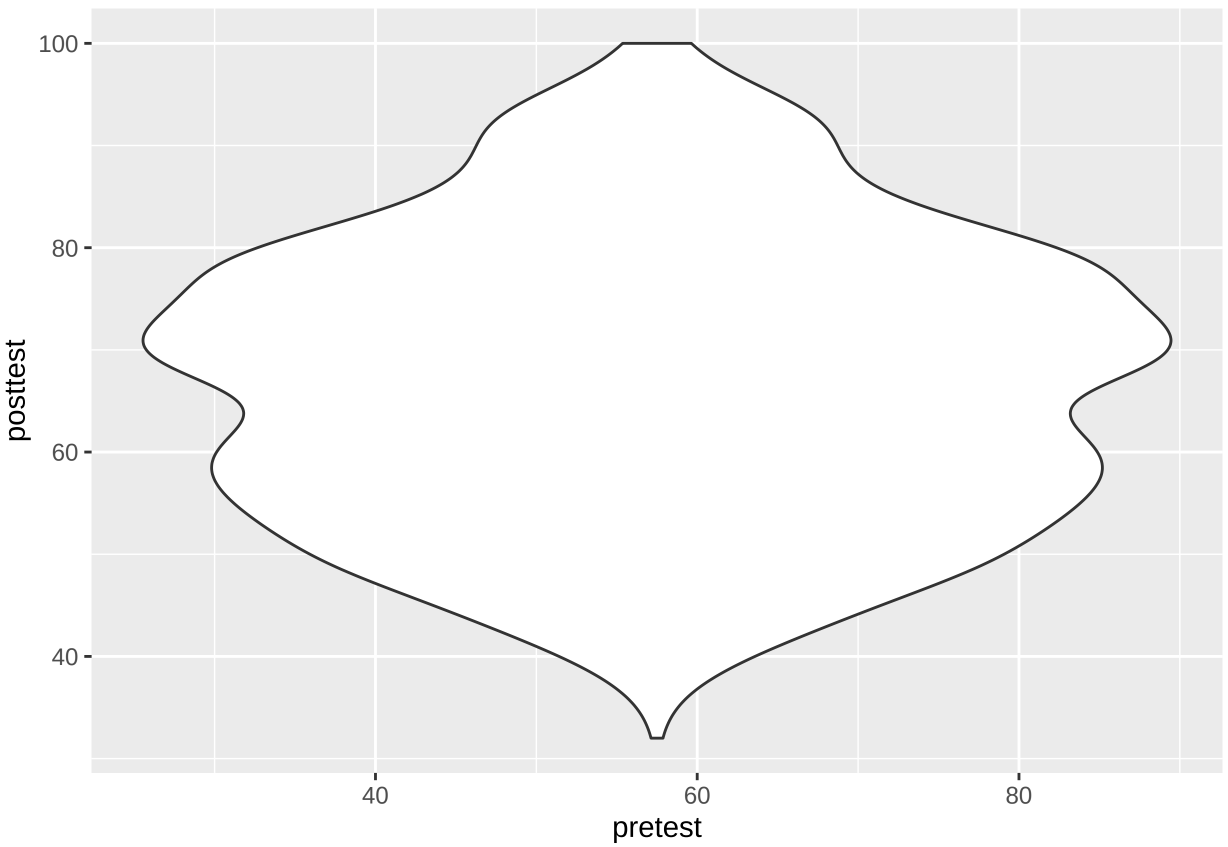
Visualization of the violin plot by adding tails at the end

```
ggplot(data,aes(x=pretest,y=posttest))+ geom_violin(trim = FALSE)
```



Visualization of the violin plot with area proportional to number of observations

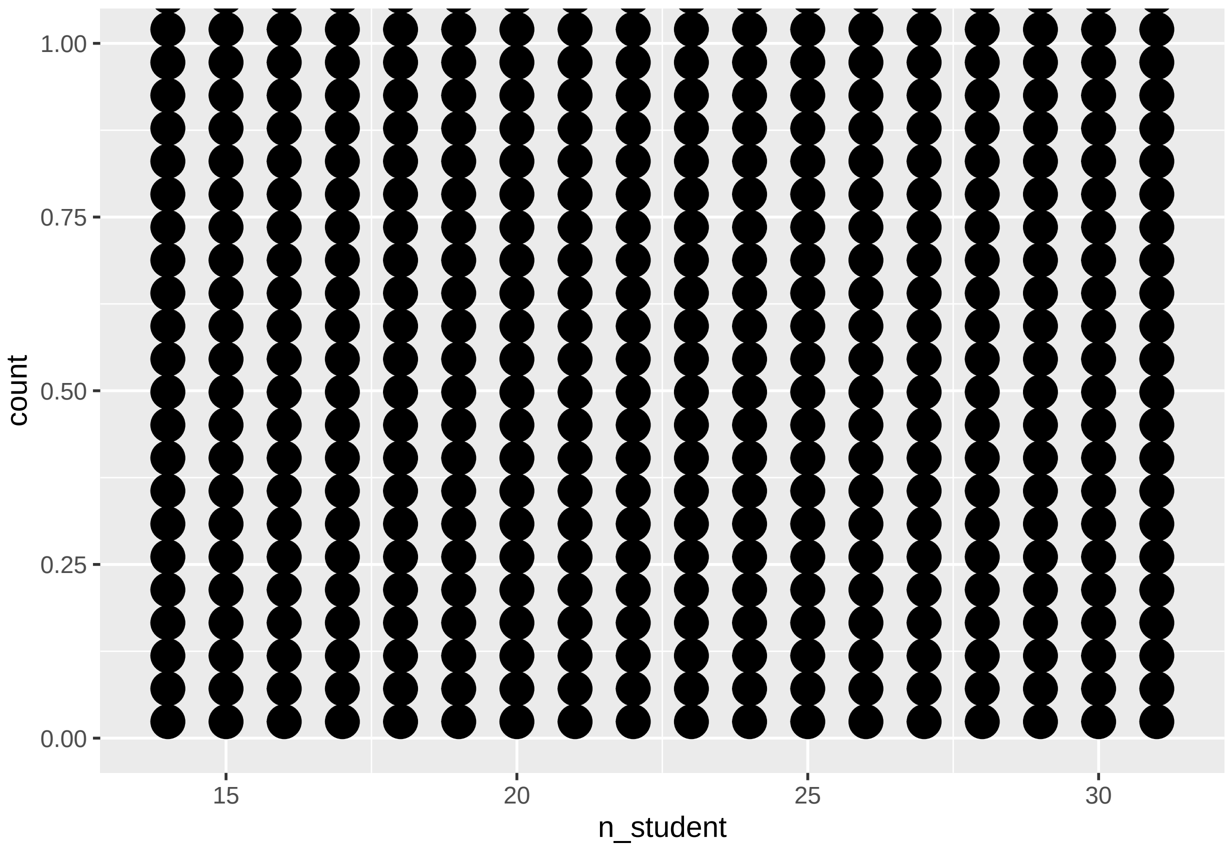
```
ggplot(data,aes(x=pretest,y=posttest)) + geom_violin(scale = "count")
```



Visualization of the dot plot using the geom_dotplot() function

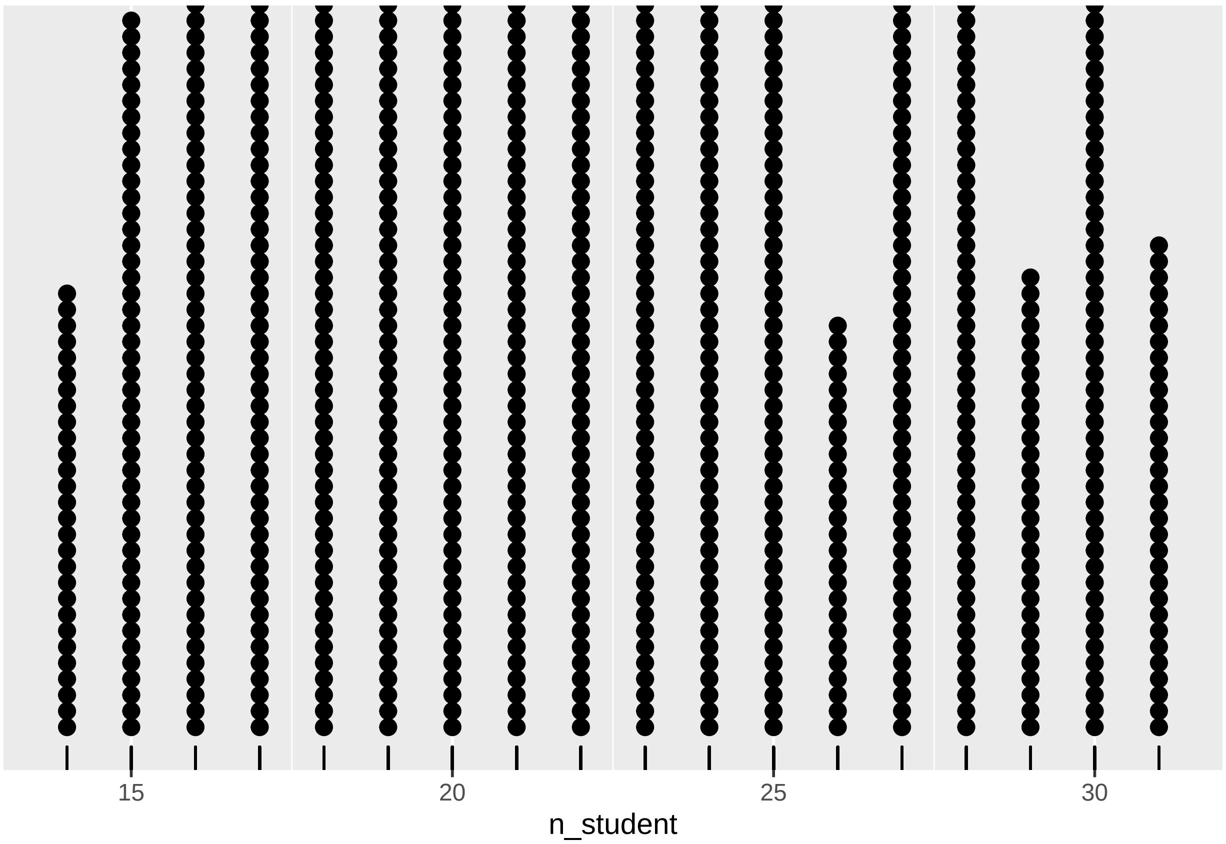
```
ggplot(data, aes(x = n_student)) +
 geom_dotplot()
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with ‘binwidth‘.
```



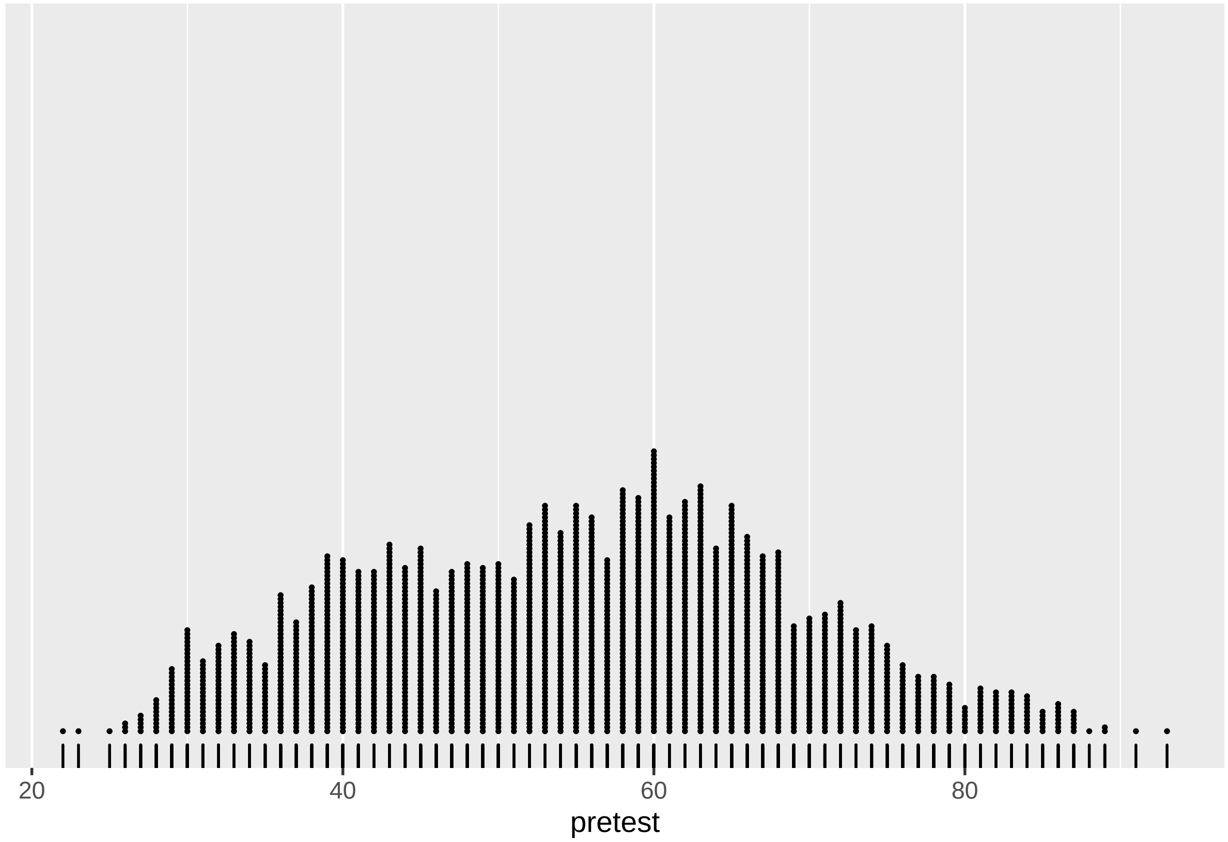
Visualization of dot plot using geom_dotplot() bins the data along the x-axis and stacks on the y-axis

```
ggplot(data,aes(x=n_student)) +
geom_dotplot(binwidth = .25) +
geom_rug() +
scale_y_continuous(breaks = NULL) +
theme(axis.title.y = element_blank())
```



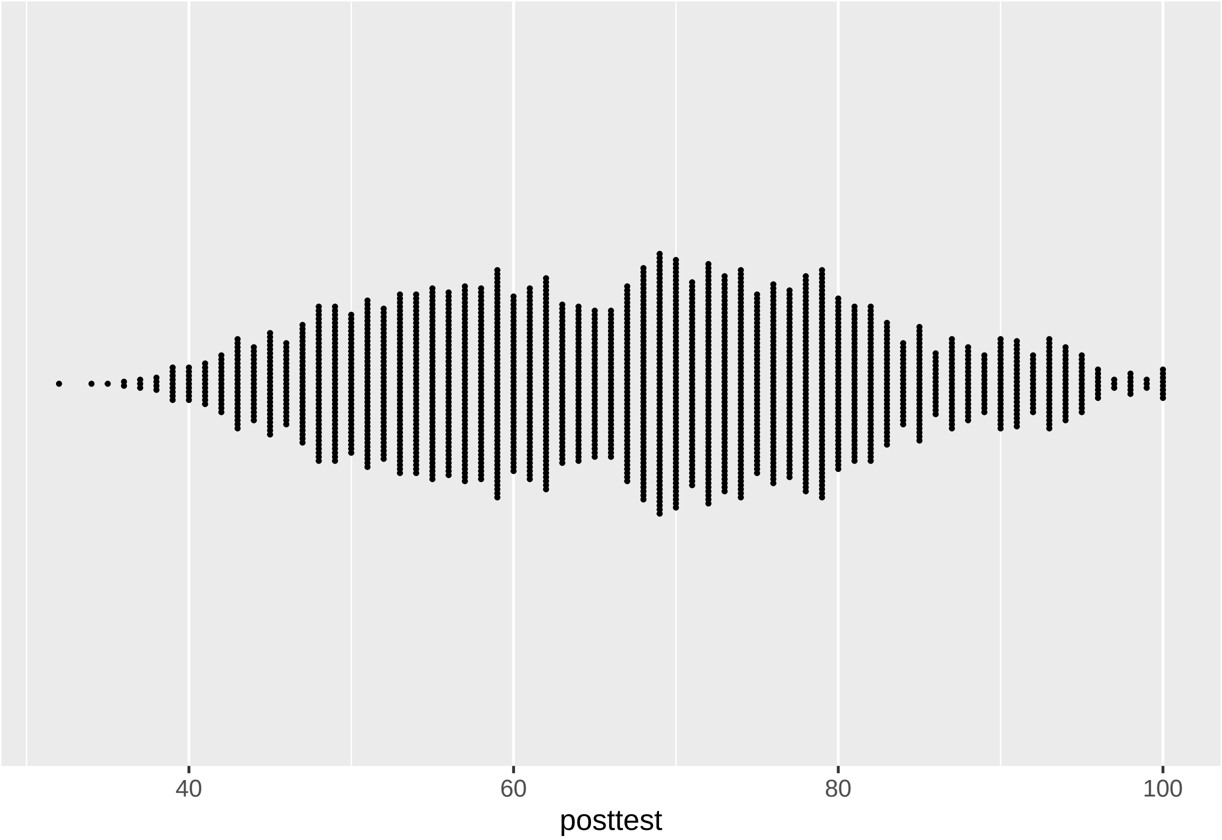
Visualization of dot plot using dotplot() function and to use bins that are arranged with a fixed, regular spacing, like a histogram.

```
ggplot(data,aes(x=pretest)) +
geom_dotplot(method = "histodot", binwidth = .25) +
geom_rug() +
scale_y_continuous(breaks = NULL) +
theme(axis.title.y = element_blank())
```



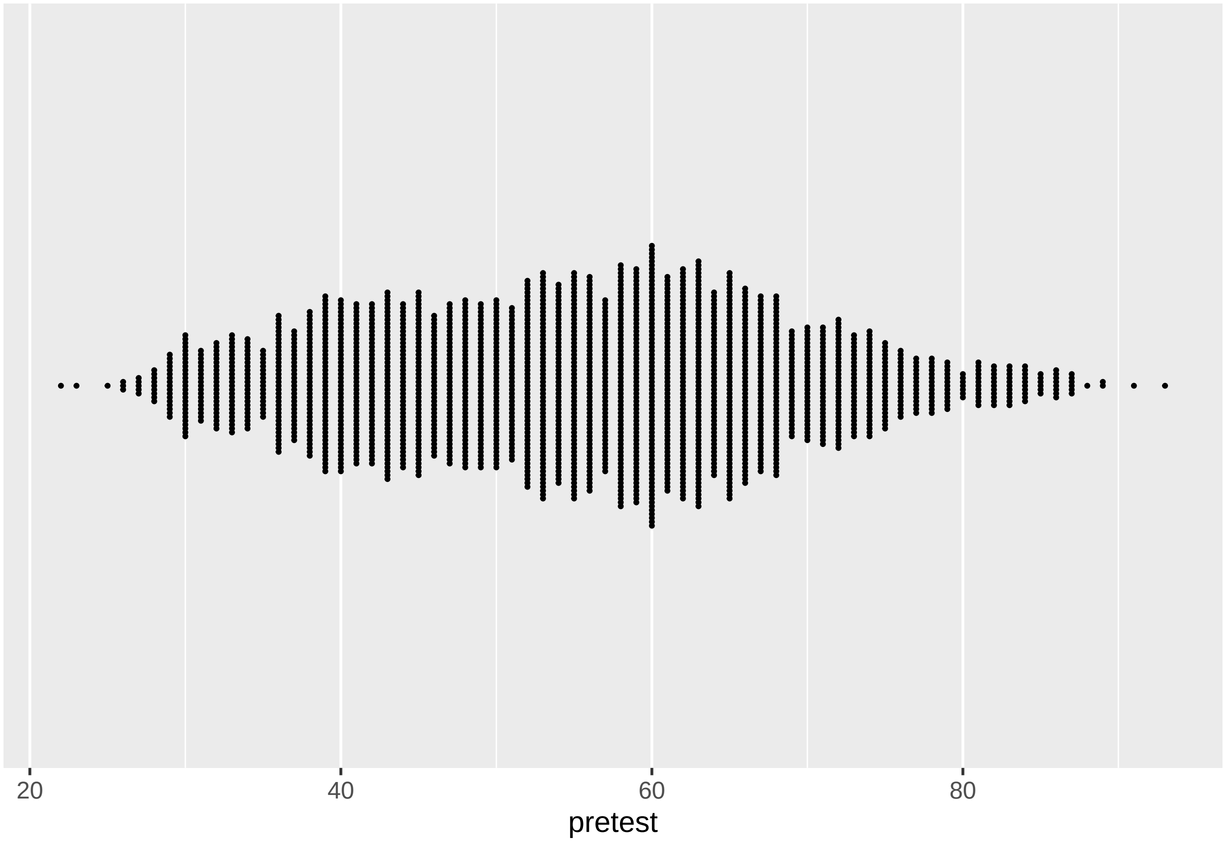
Visualization of dot plot and the dots can also be stacked centered using stackdir="center"

```
ggplot(data,aes(x=posttest))+
geom_dotplot(binwidth = .25, stackdir = "center") +
scale_y_continuous(breaks = NULL) +
theme(axis.title.y = element_blank())
```



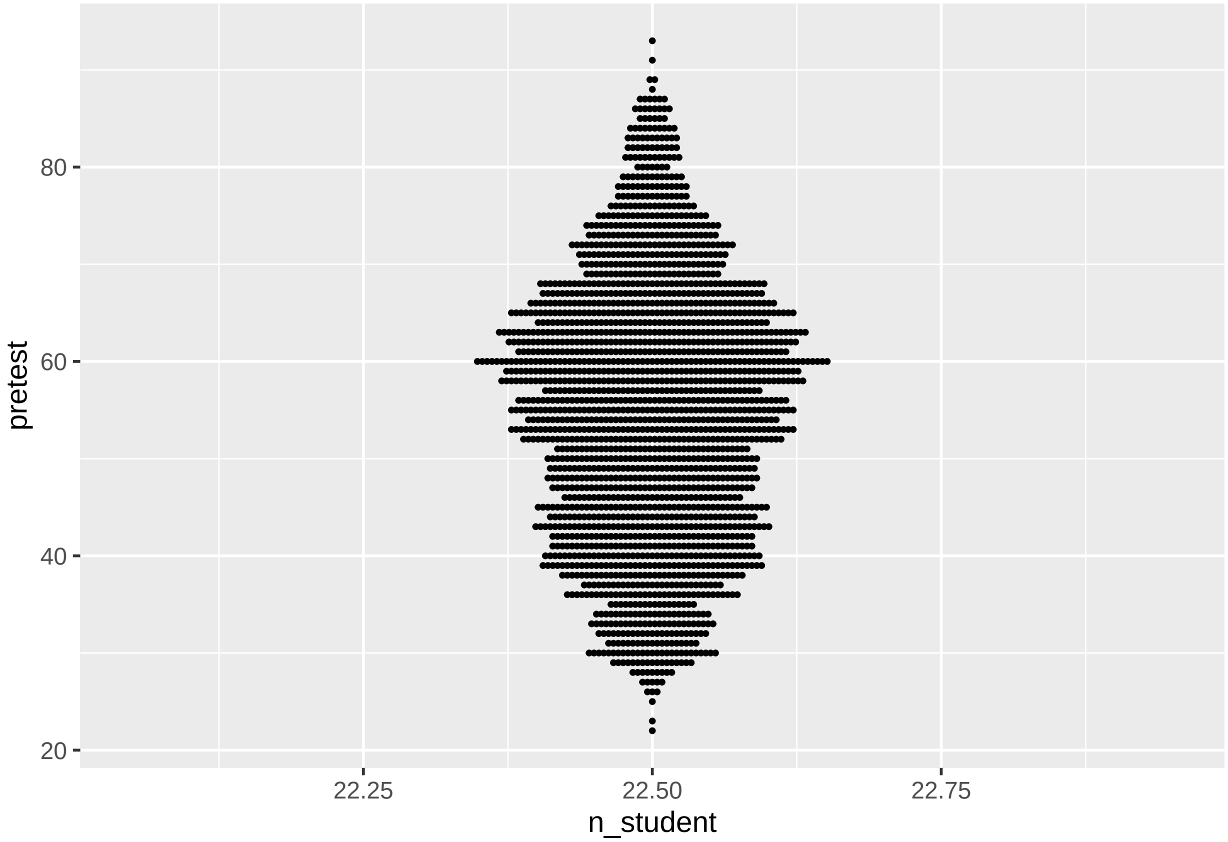
Visualization of dot plot and the dots can also be stacked centered using stackdir="centerwhole"

```
 ggplot(data,aes(x=pretest)) +
geom_dotplot(binwidth = .25, stackdir = "centerwhole") +
scale_y_continuous(breaks = NULL) +
theme(axis.title.y = element_blank())
```



Visualization of dotplot and to stack the dots along the y-axis, and group them along the x-axis, by setting binaxis = “y”.

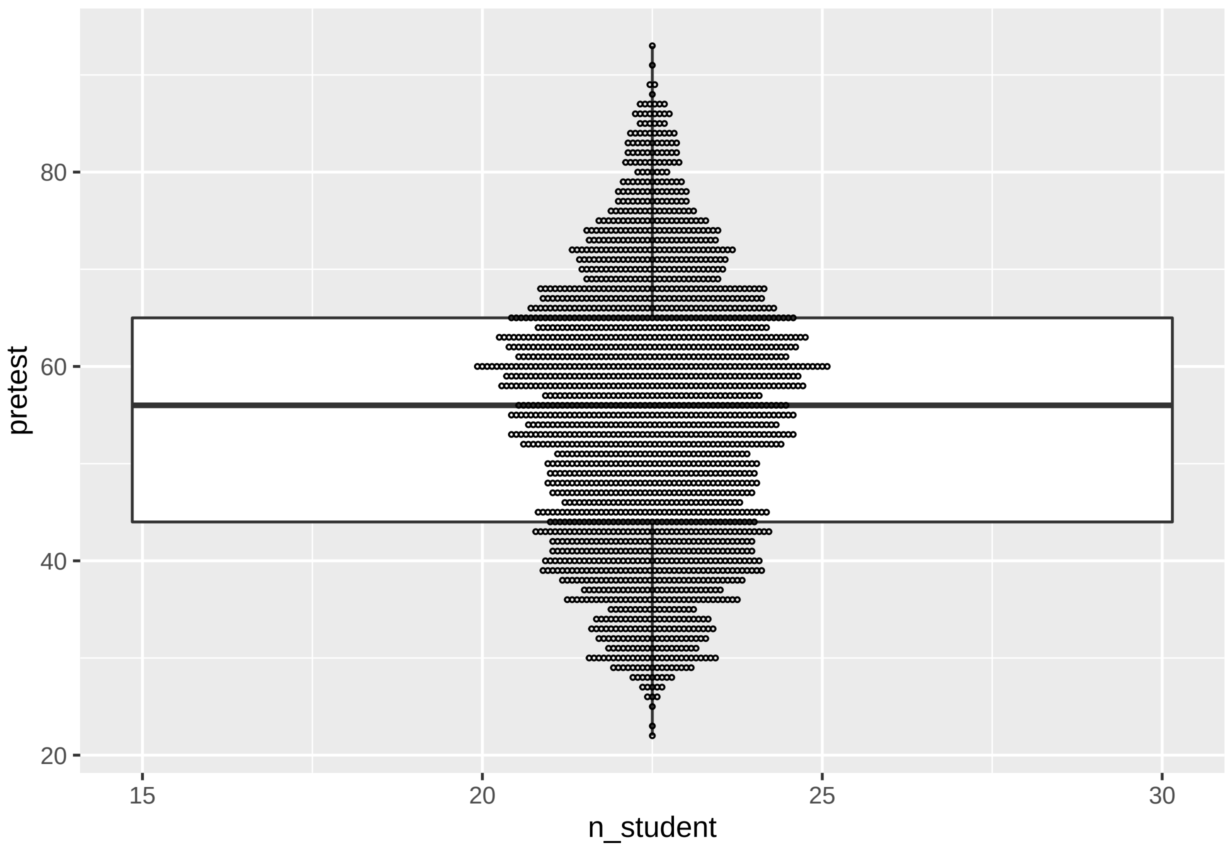
```
ggplot(data, aes(x = n_student, y = pretest)) +
 geom_dotplot(binaxis = "y", binwidth = .5, stackdir = "center")
```



Visualization of dot plots overlaid on box plots and to make the dots hollow and have the box plots not show outliers

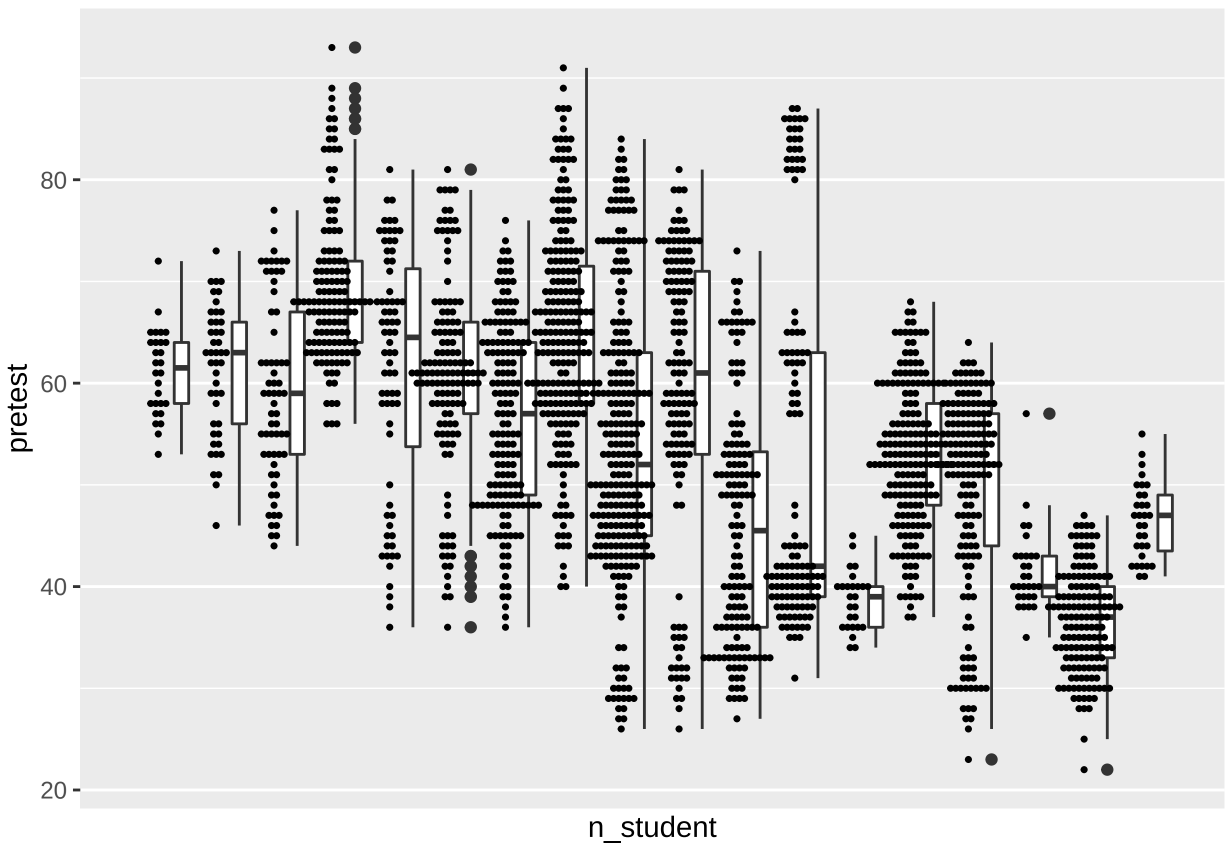
```
ggplot(data, aes(x = n_student, y = pretest)) +
 geom_boxplot(outlier.colour = NA, width = .4) +
 geom_dotplot(binaxis = "y", binwidth = .5, stackdir = "center", fill = NA)
```

```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



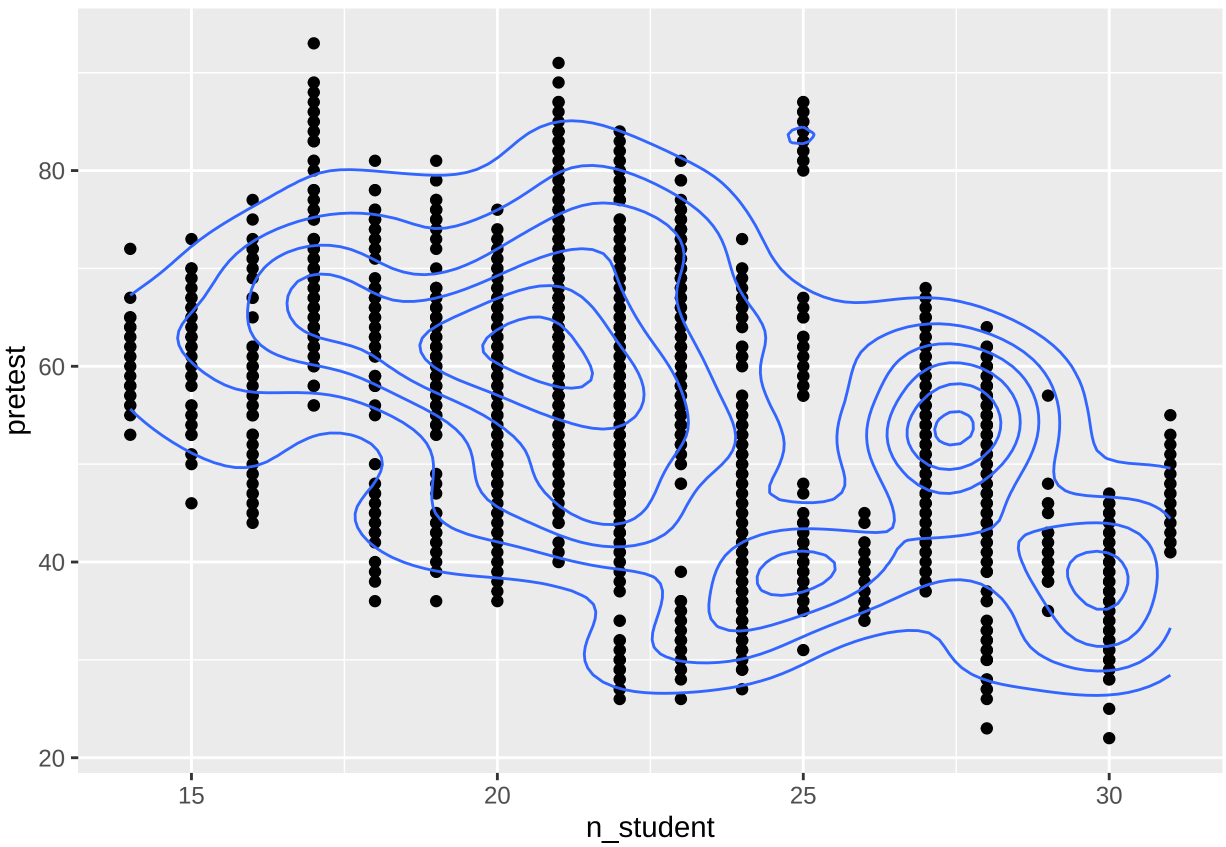
Visualization of dot plot on top of box plot by specifying binaxis,binwidth,stackdir

```
ggplot(data, aes(x = n_student, y = pretest)) +
 geom_boxplot(aes(x = as.numeric(n_student) + .2, group = n_student), width = .25) +
 geom_dotplot(
 aes(x = as.numeric(n_student) - .2, group = n_student),
 binaxis = "y",
 binwidth = .5,
 stackdir = "center"
 ) +
 scale_x_continuous(
 breaks = 1:nlevels(data$pretest),
 labels = levels(data$pretest)
 )
```



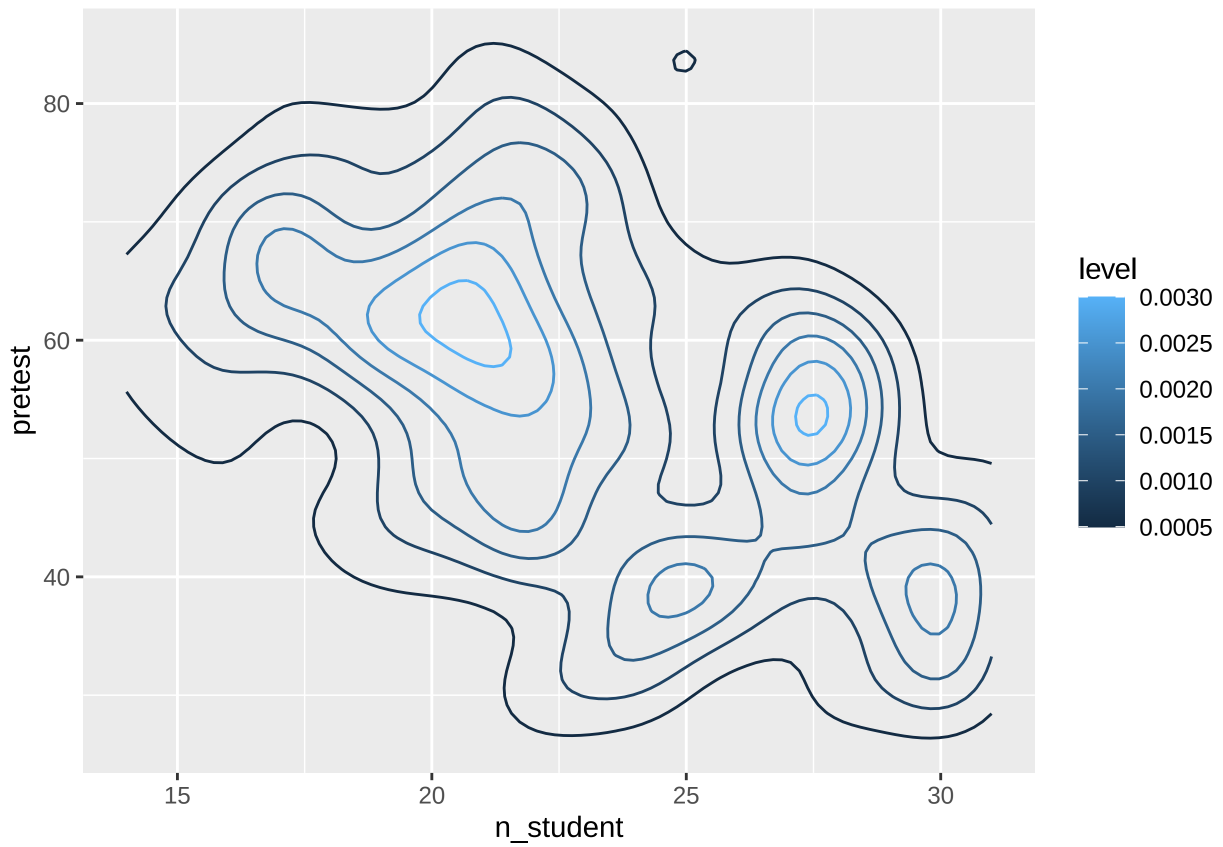
Plotting the density of two-dimensional data

```
ggplot(data,aes(x=n_student,y=pretest)) +
geom_point() +
stat_density2d()
```



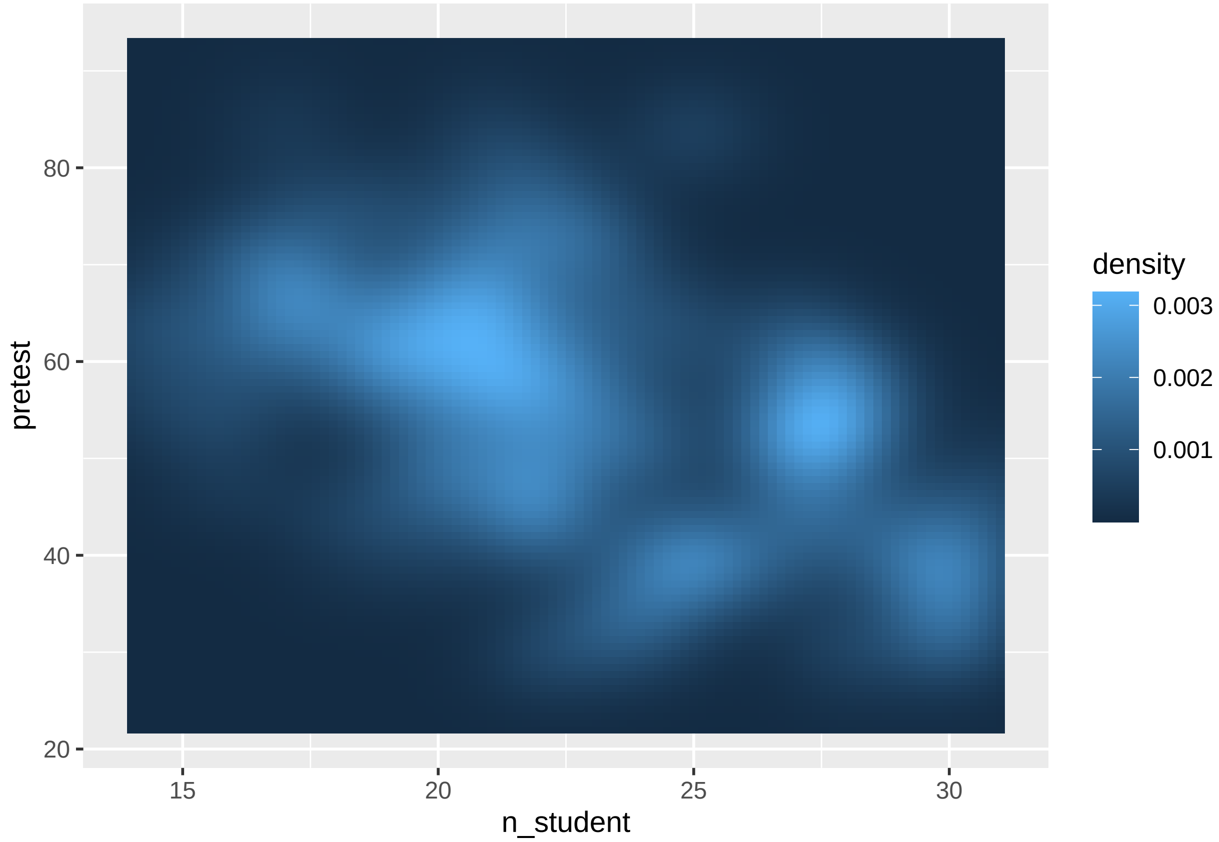
Maps the height of the density curve to the color of the contour lines, by using ..level..

```
ggplot(data,aes(x=n_student,y=pretest))+
 stat_density2d(aes(colour = ..level..))
```



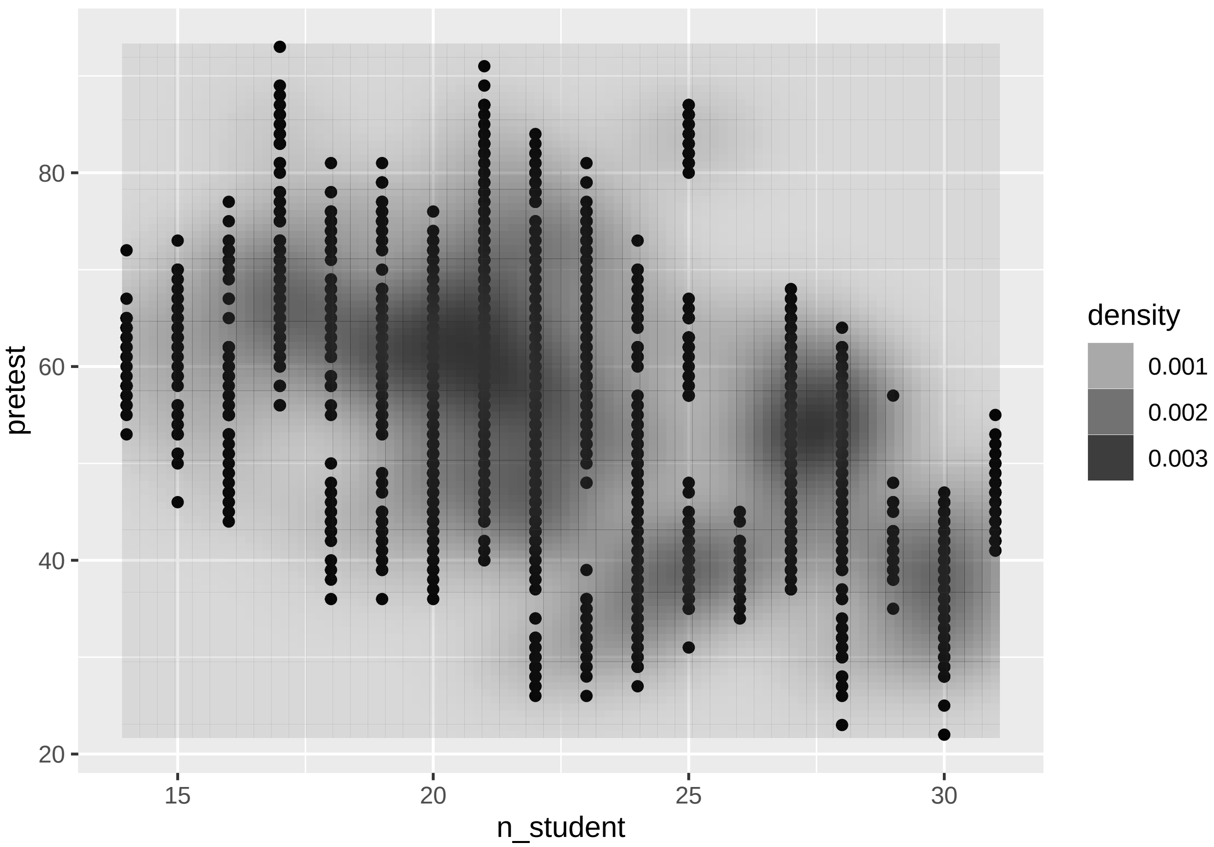
Maps density estimate to fill color

```
ggplot(data,aes(x=n_student,y=pretest)) +
 stat_density2d(aes(fill = ..density..), geom = "raster", contour = FALSE)
```



With points, and map density estimate to alpha

```
ggplot(data,aes(x=n_student,y=pretest)) +
geom_point() +
stat_density2d(aes(alpha = ..density..), geom = "tile", contour = FALSE)
```


pretest
80
60
40
20
15
20
25
30
n_student
density
0.001
0.002
0.003