EDA_LA_2

Reshma Itagi

2022-09-23

```
df = read.csv('test_scores.csv')
head(df)
```

```
##   school school_setting school_type classroom teaching_method n_student
## 1  ANKYI          Urban  Non-public       6OL        Standard        20
## 2  ANKYI          Urban  Non-public       6OL        Standard        20
## 3  ANKYI          Urban  Non-public       6OL        Standard        20
## 4  ANKYI          Urban  Non-public       6OL        Standard        20
## 5  ANKYI          Urban  Non-public       6OL        Standard        20
## 6  ANKYI          Urban  Non-public       6OL        Standard        20
##   student_id gender            lunch pretest posttest
## 1      2FHT3 Female Does not qualify      62       72
## 2      3JIVH Female Does not qualify      66       79
## 3      3XOWE   Male Does not qualify      64       76
## 4      556O0 Female Does not qualify      61       77
## 5      74LOE   Male Does not qualify      64       76
## 6      7YZO8 Female Does not qualify      66       74
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

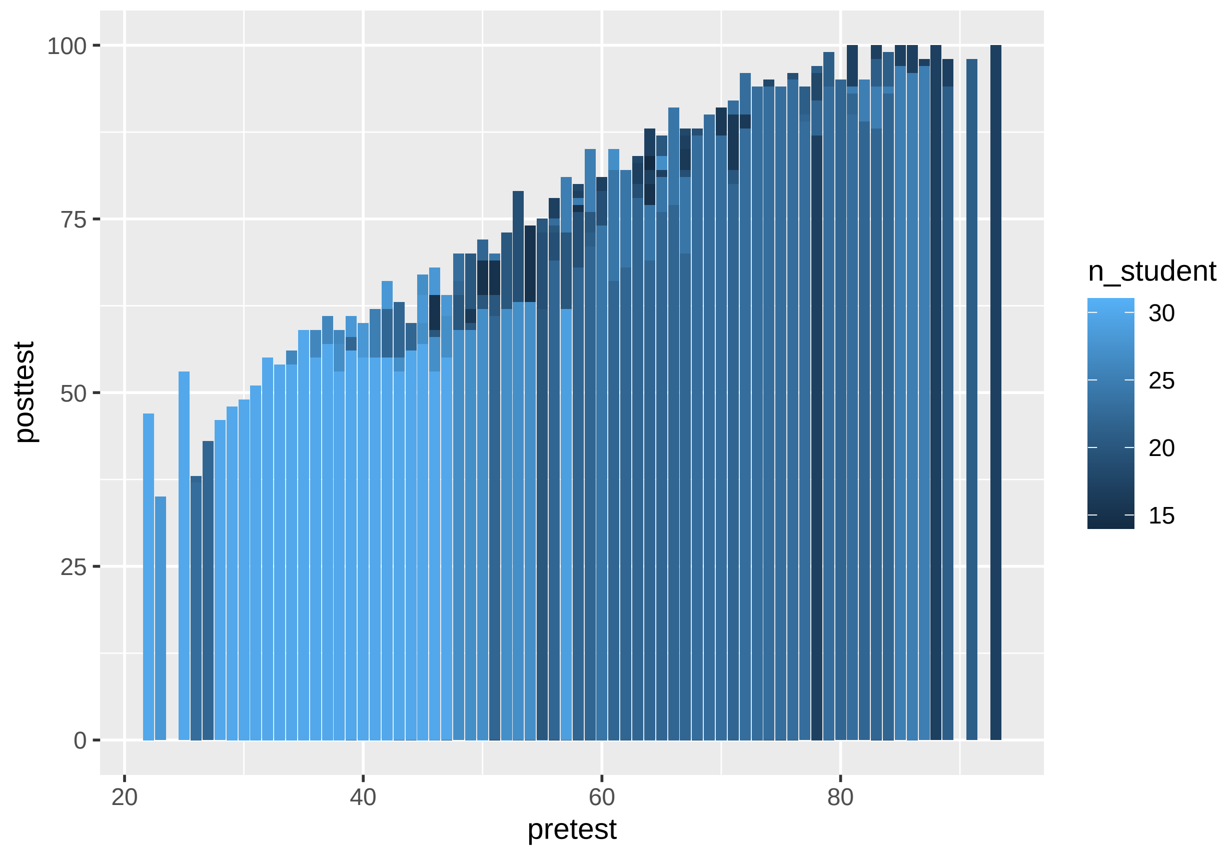
```
library(ggplot2)
```

```
temp = df %>% filter(df$school == "ANKYI")
temp.aov = na.omit(aov(temp$pretest ~ temp$posttest, data = temp))
temp.aov
```

```
## Call:
##    aov(formula = temp$pretest ~ temp$posttest, data = temp)
```

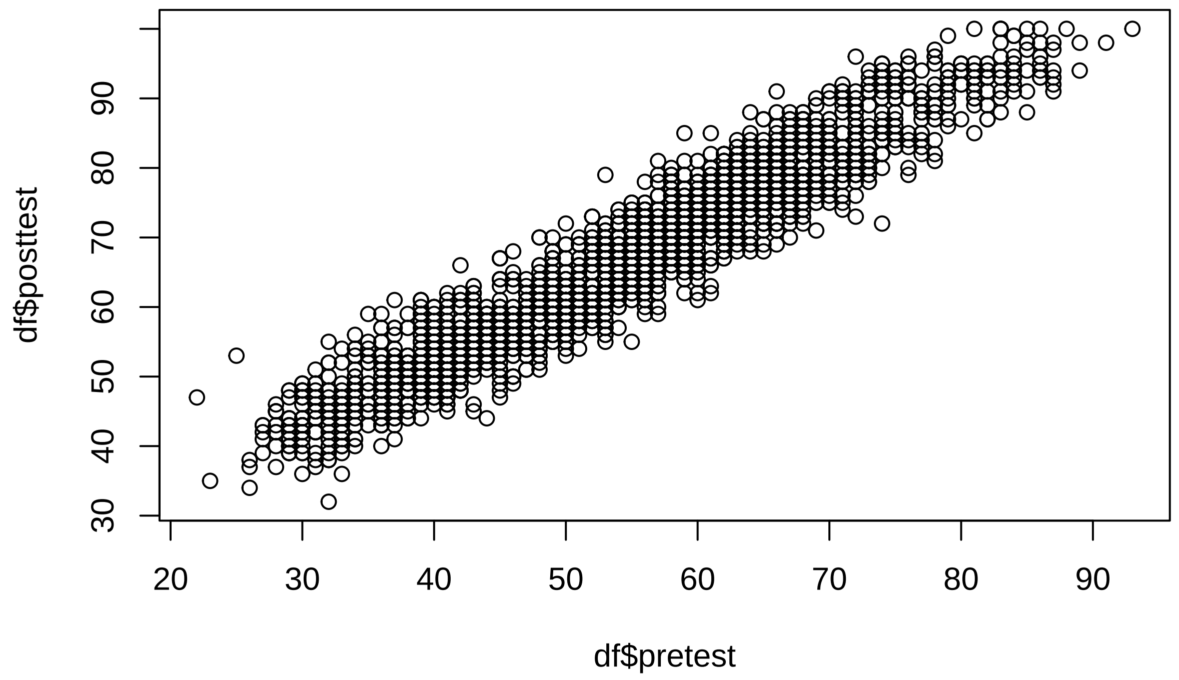
```
##
## Terms:
##                 temp$posttest Residuals
## Sum of Squares       200.8709  168.3486
## Deg. of Freedom             1        39
##
## Residual standard error: 2.07765
## Estimated effects may be unbalanced
```

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



Creating a Scatter Plot

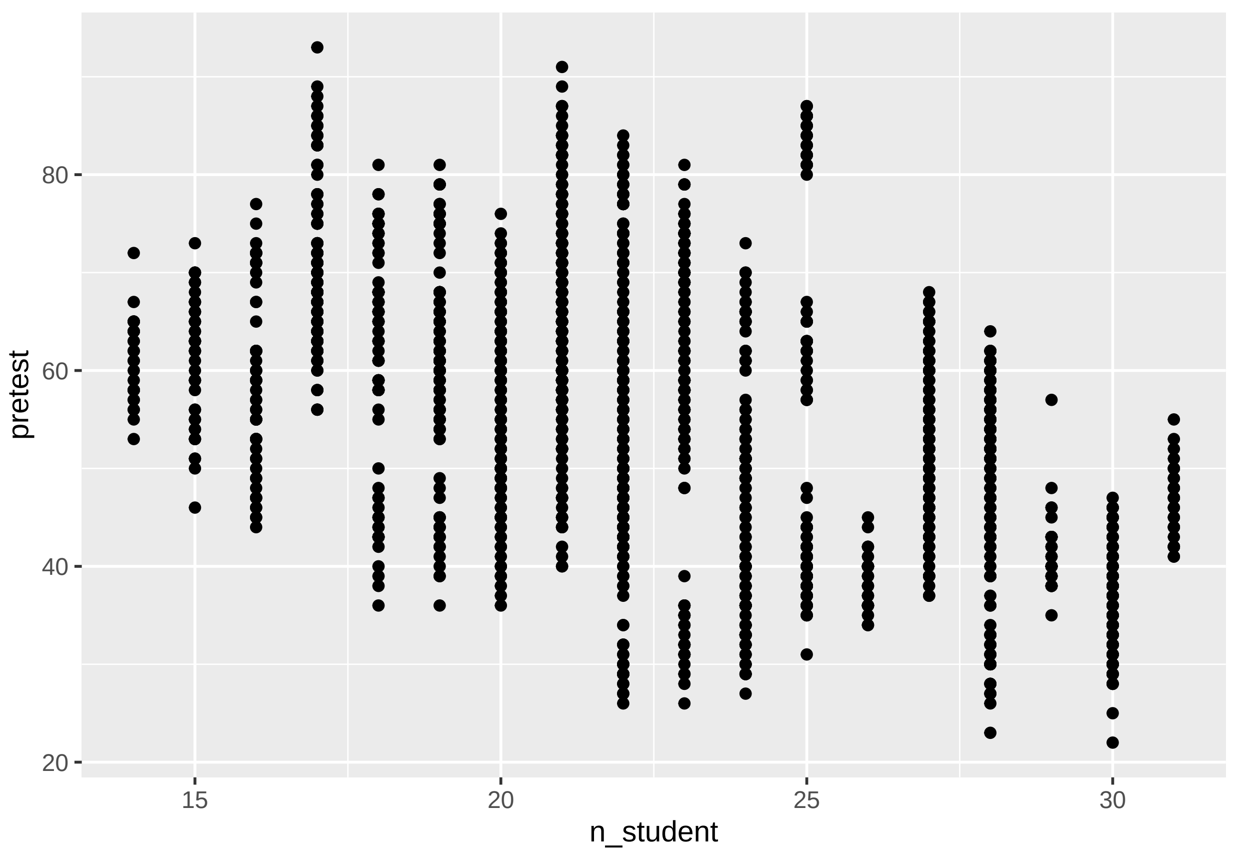
```
plot(df$pretest, df$posttest)
```



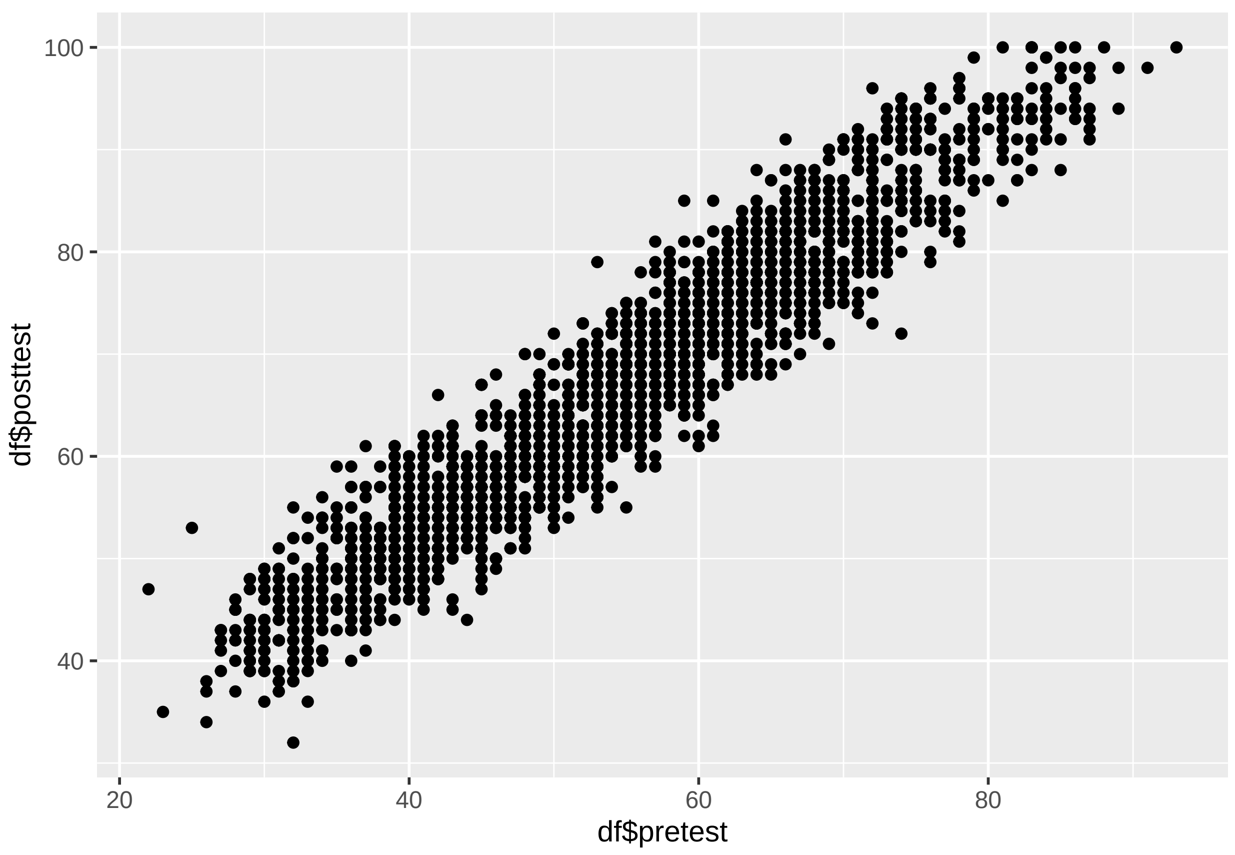
```
# Scatter plot with base graphics
```

Scatter plot with ggplot 2

```
library(ggplot2)
ggplot(df, aes(x = n_student, y = pretest)) +geom_point()
```

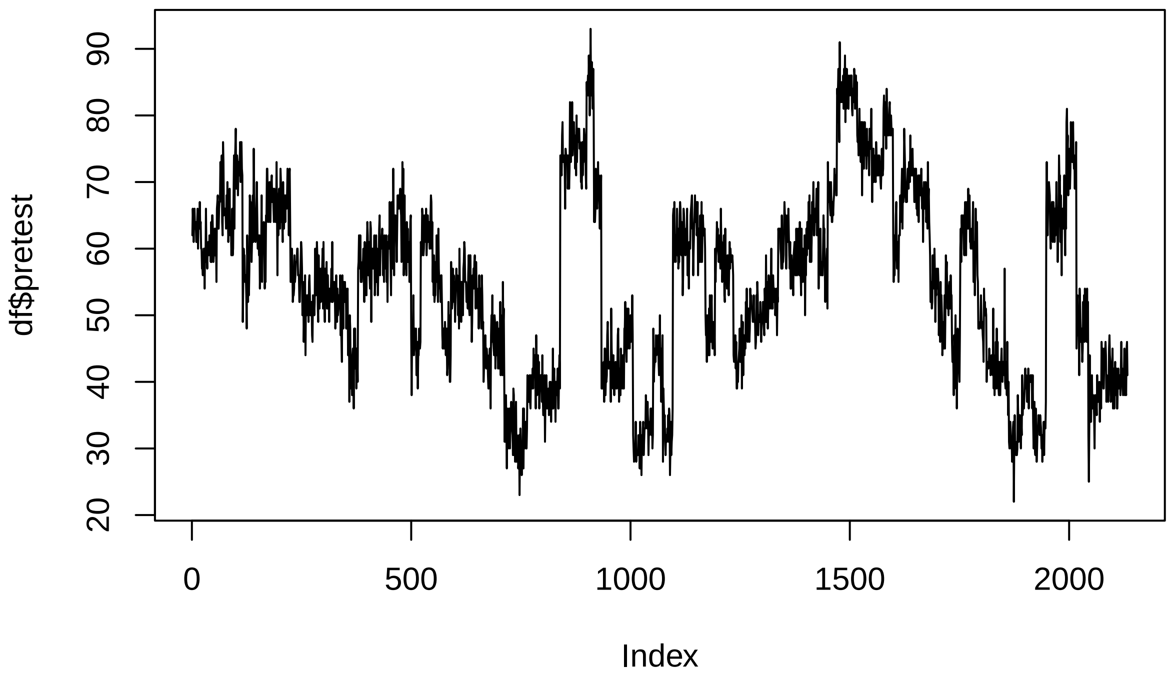


```
ggplot(data = NULL, aes(x =df$pretest, y=df$posttest )) +
geom_point()
```



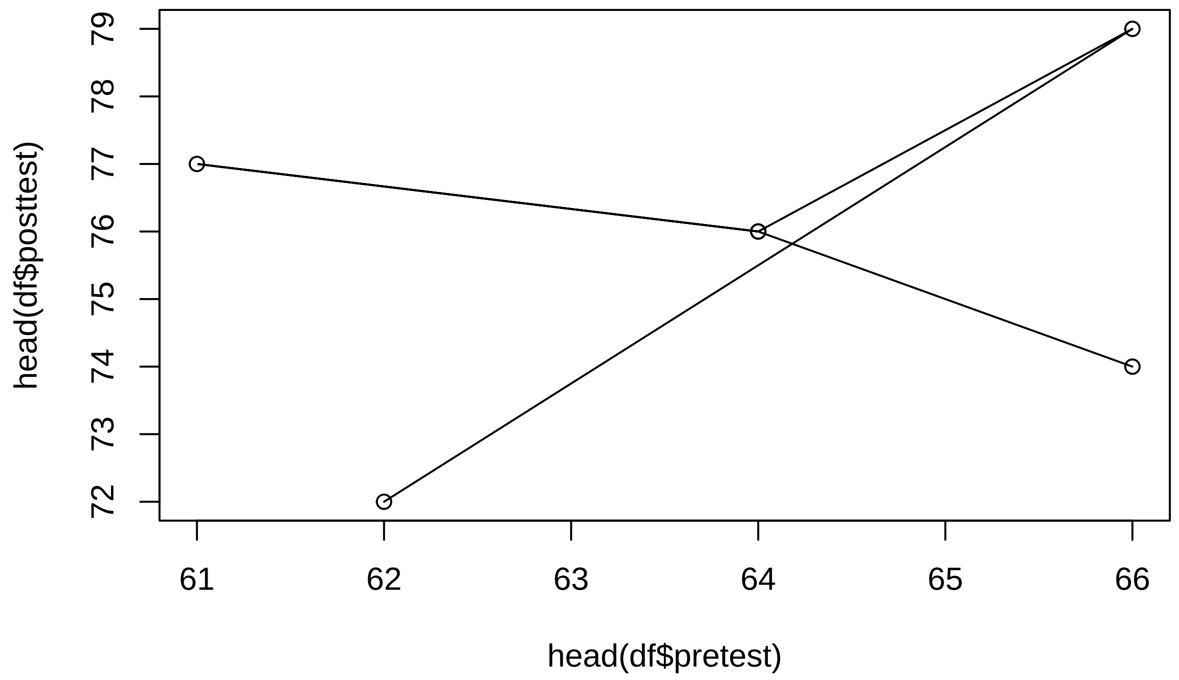
Creating a Line Graph

```
plot(df$pretest,type = "l")
```



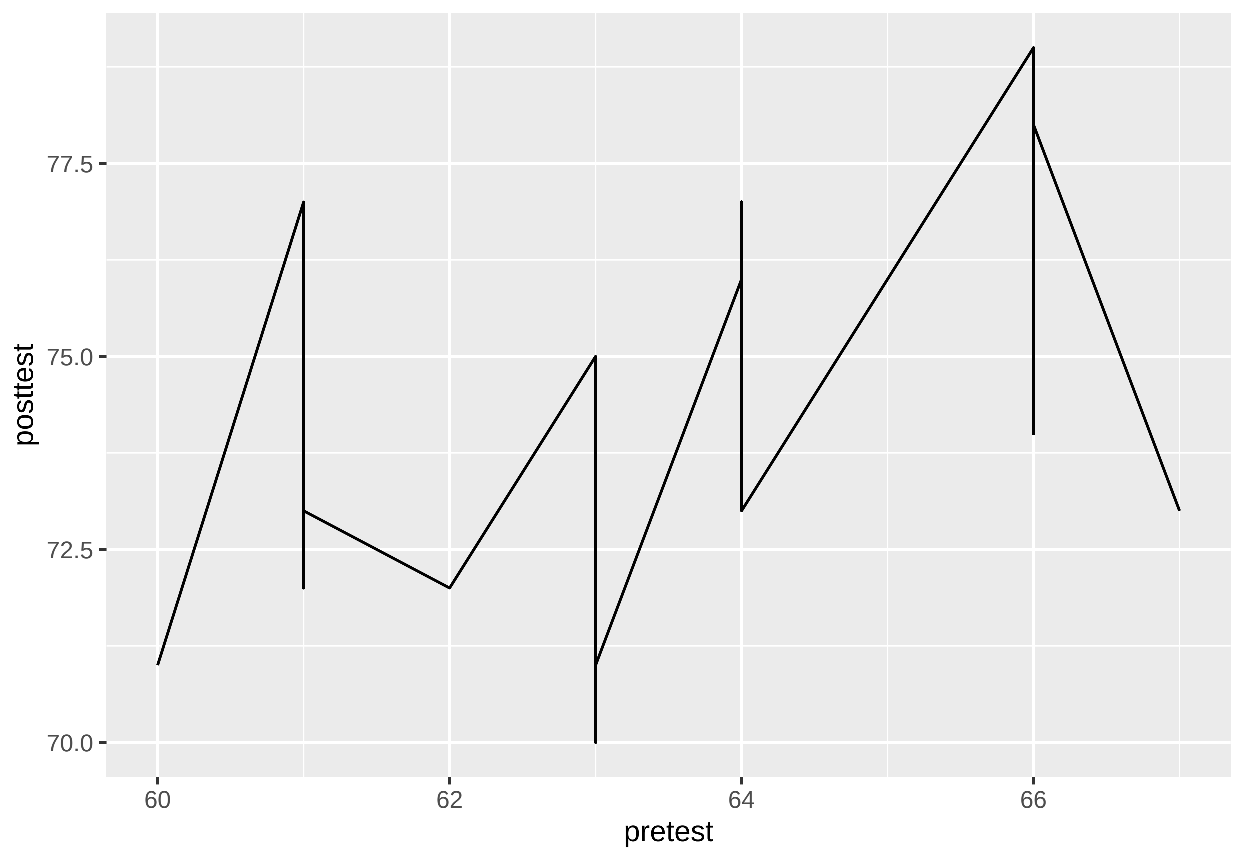
Line graph with base graphics

```
plot(head(df$pretest),head(df$posttest),  type = "l")
points(head(df$pretest),head(df$posttest))
```

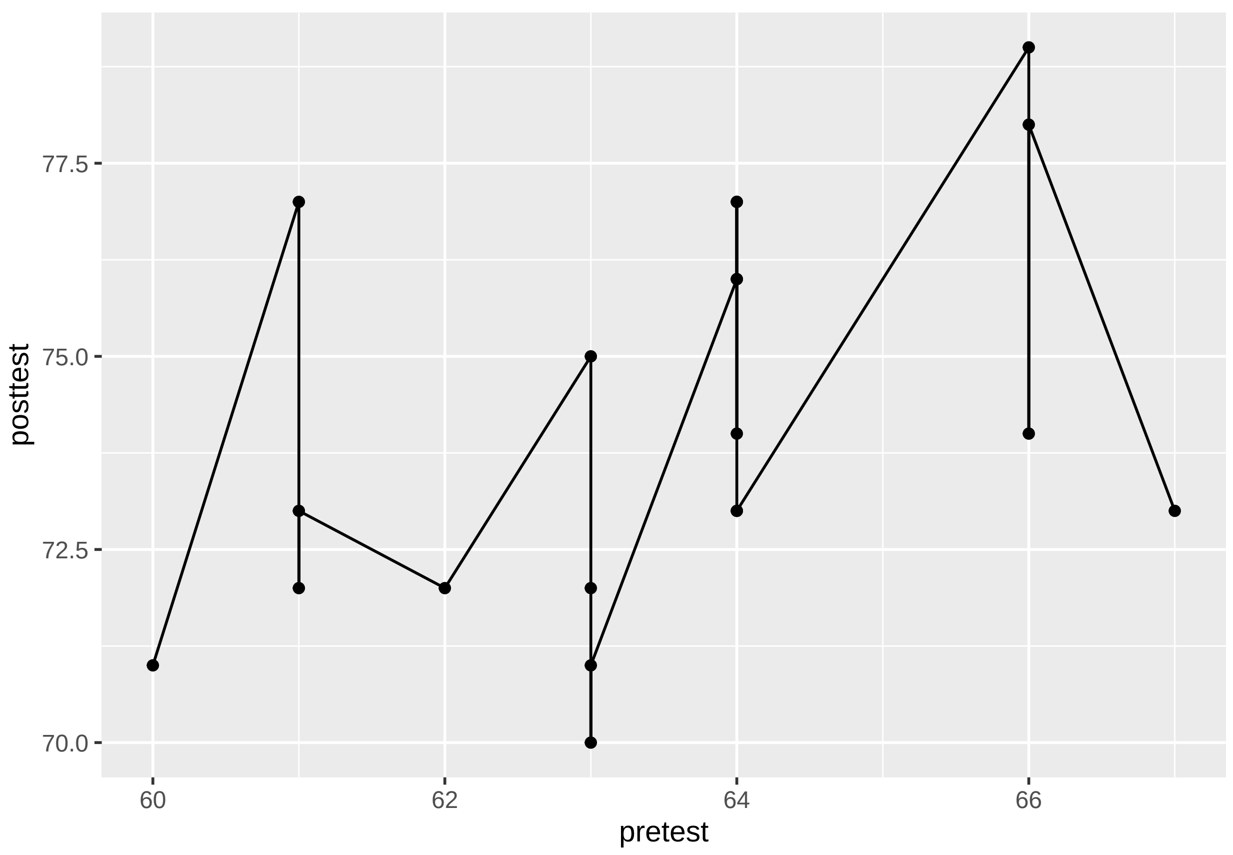


# Line graph with ggplot() and With points added to ggplot()

```
ggplot(head(df,n=20), aes(x = pretest, y = posttest)) +geom_line()
```

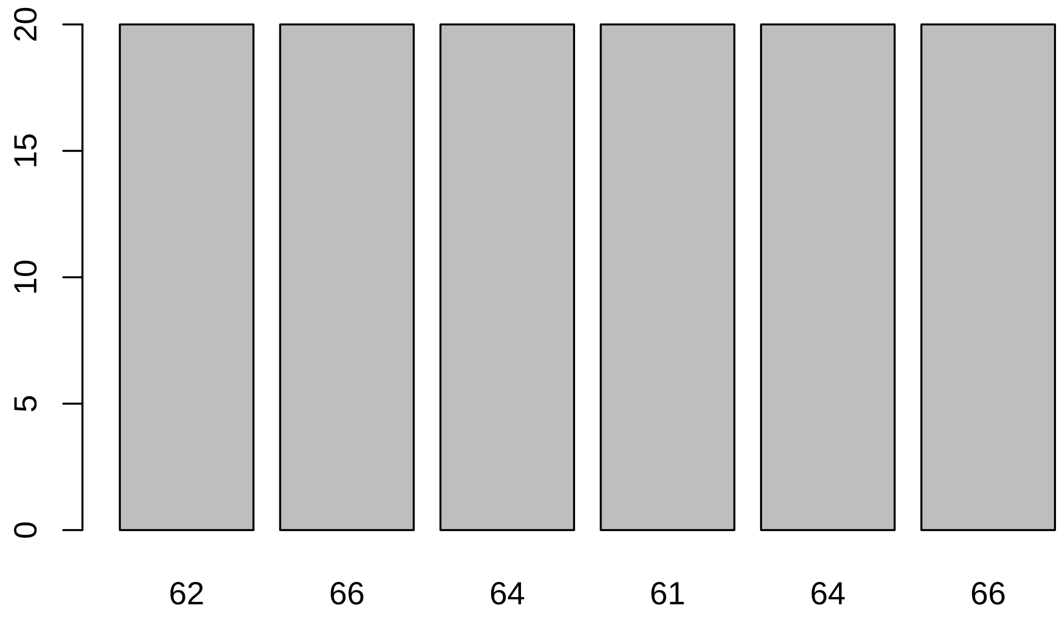


```
ggplot(head(df,n=20), aes(x = pretest, y = posttest)) +geom_line()+geom_point()
```



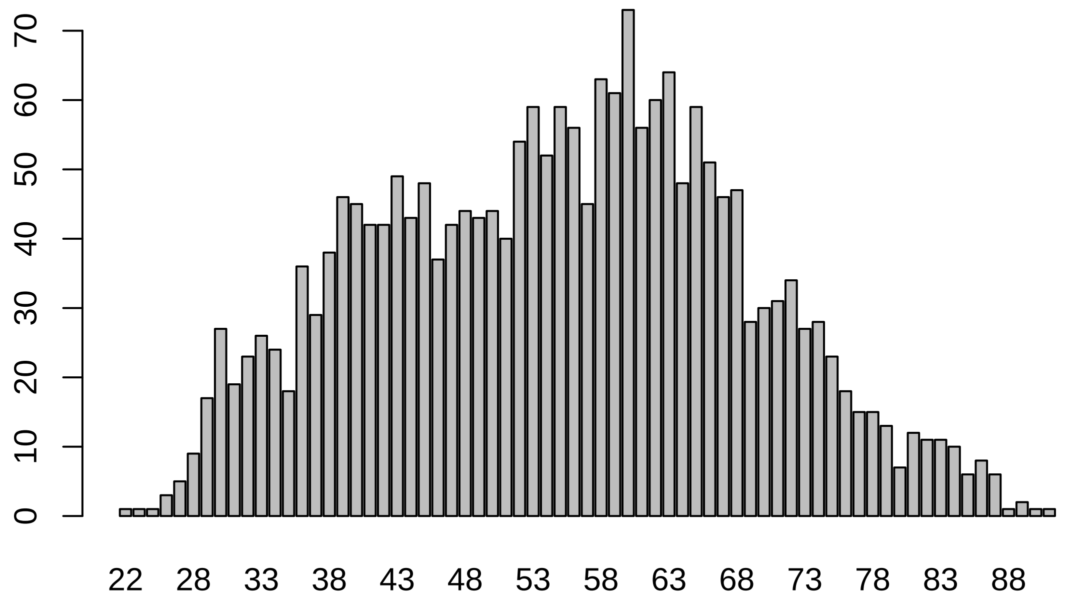
# Creating a Bar Graph

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



# Generate a table of counts

```
barplot(table(df$pretest))
```

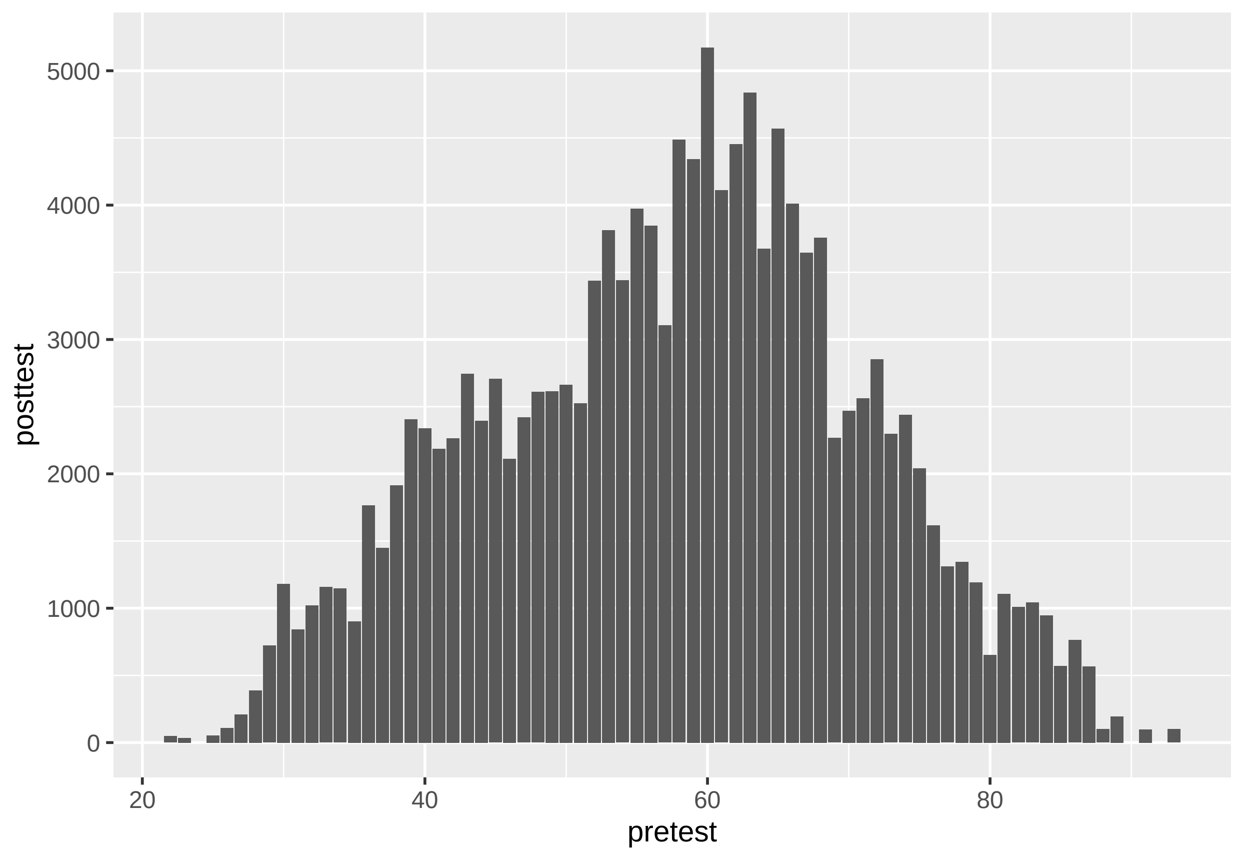


# Laoding ggplot2() package

```
library(ggplot2)
```

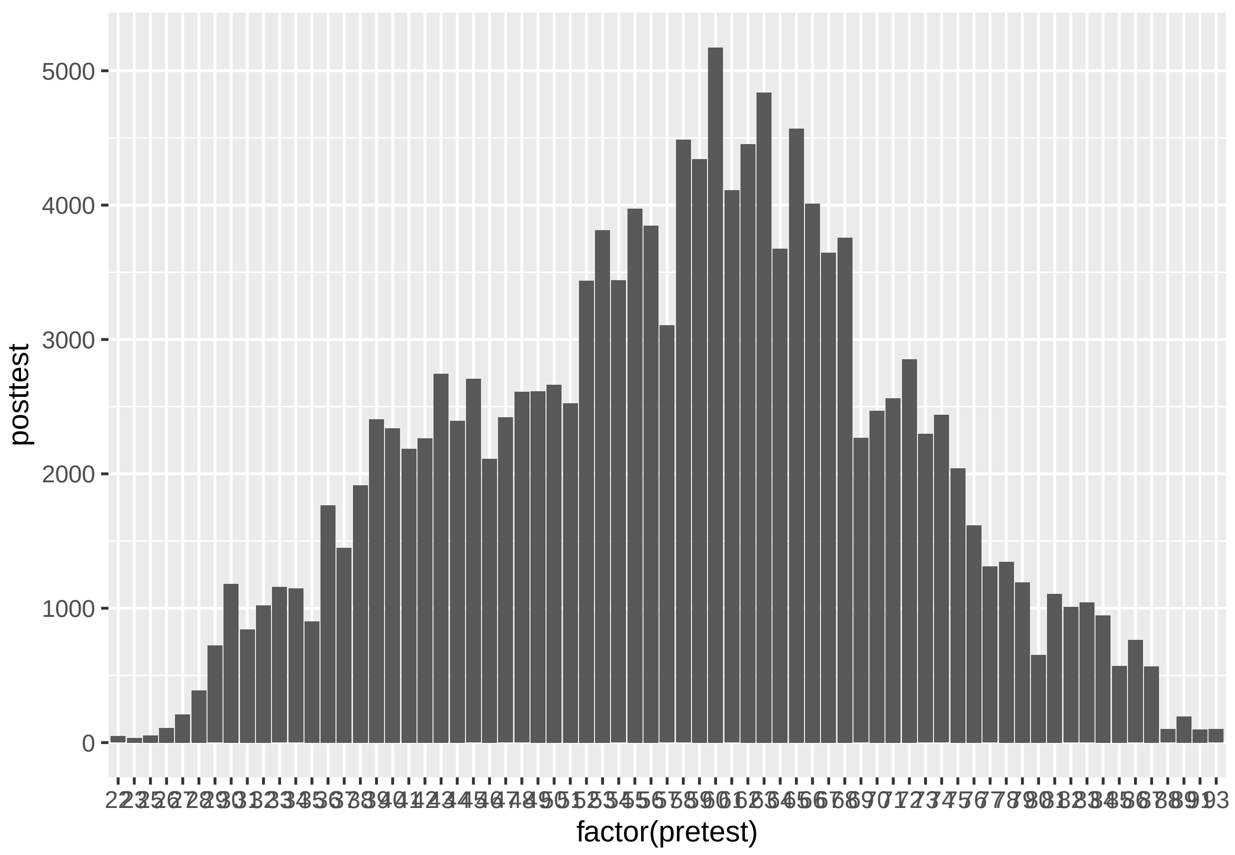
Bar graph of values. This uses the dataset data frame, with the "pretest" column for x values and the "posttest" column for y values.

```
ggplot(df, aes(x = pretest, y = posttest)) +geom_col()
```



# Convert the x variable to a factor, so that it is treated as discrete

```
ggplot(df, aes(x = factor(pretest), y = posttest)) +geom_col()
```

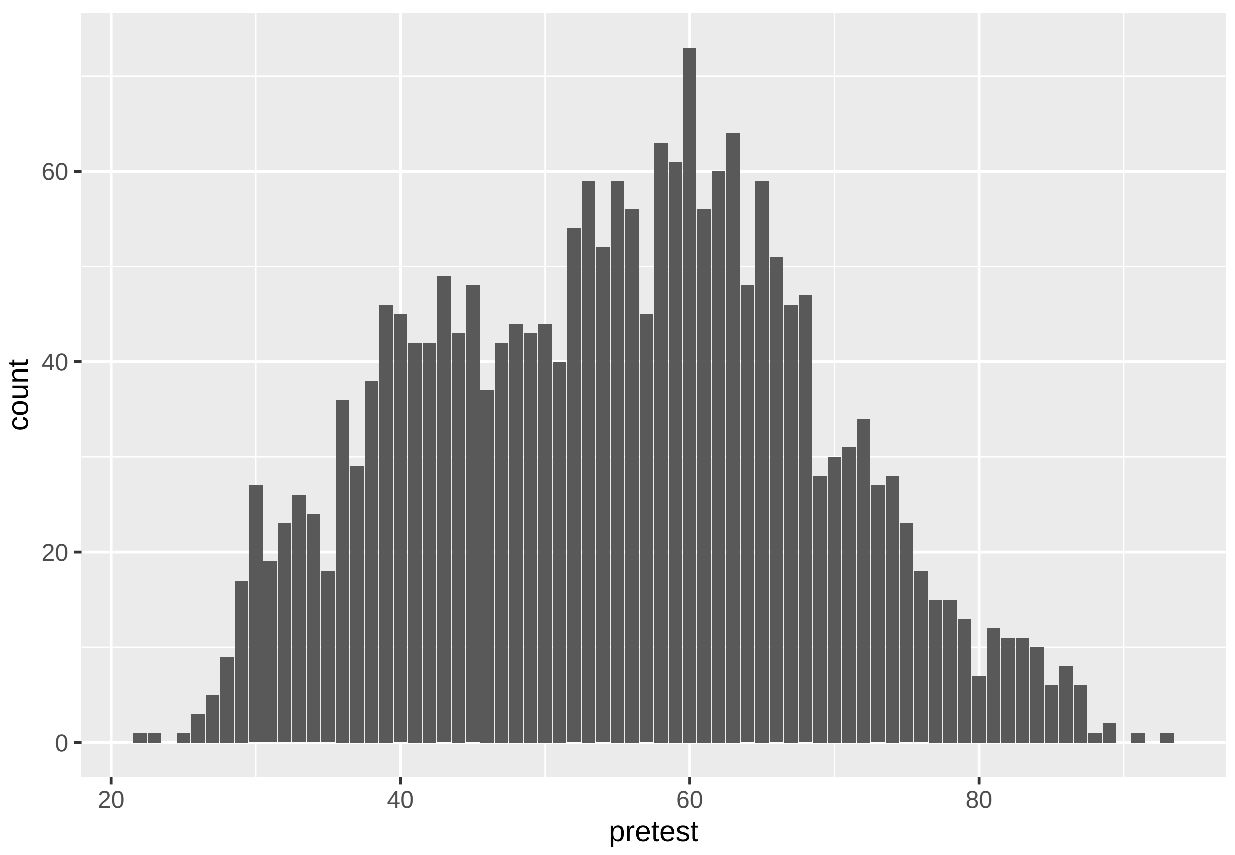


Bar graph of counts. This uses the dataset data frame, with the "pretest" column for

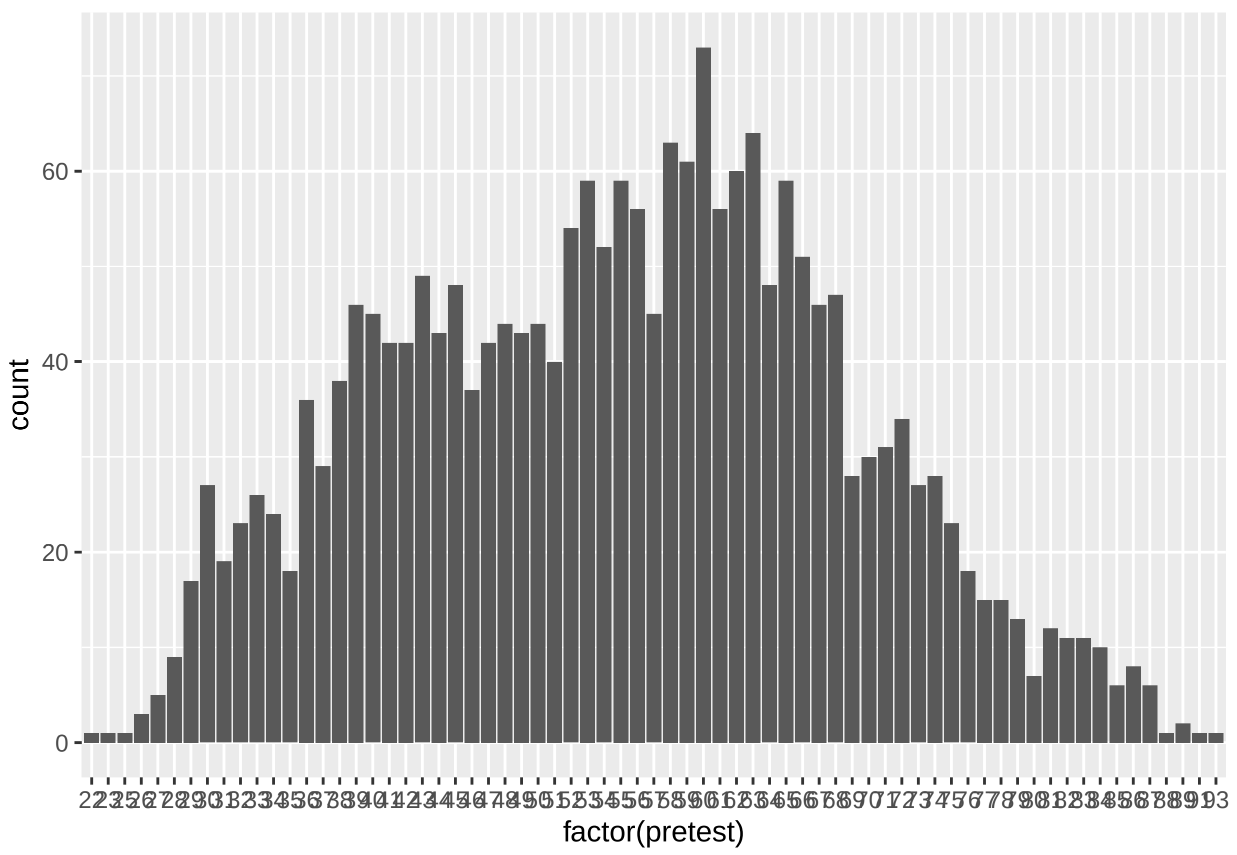
x position. The y position is calculated by counting the number of rows for

each value of pretest.

```
ggplot(df, aes(x = pretest)) +
geom_bar()
```

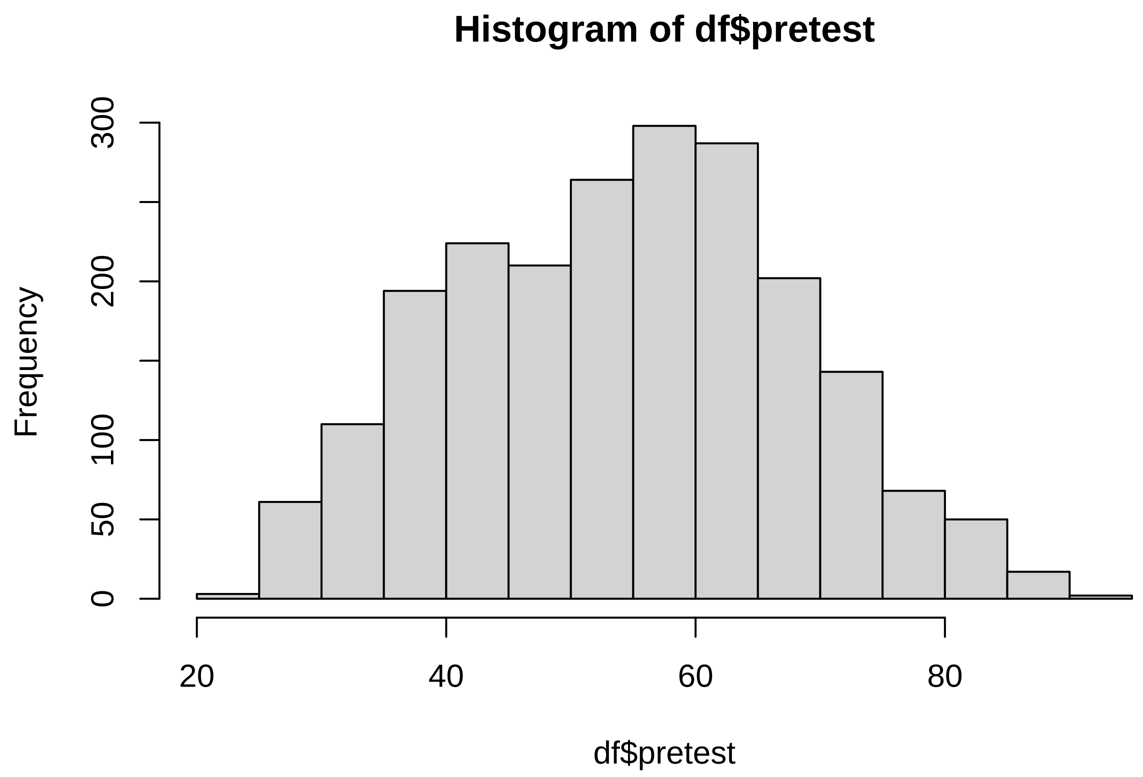


```r
# Bar graph of counts
ggplot(df, aes(x = factor(pretest))) +
geom_bar()
```

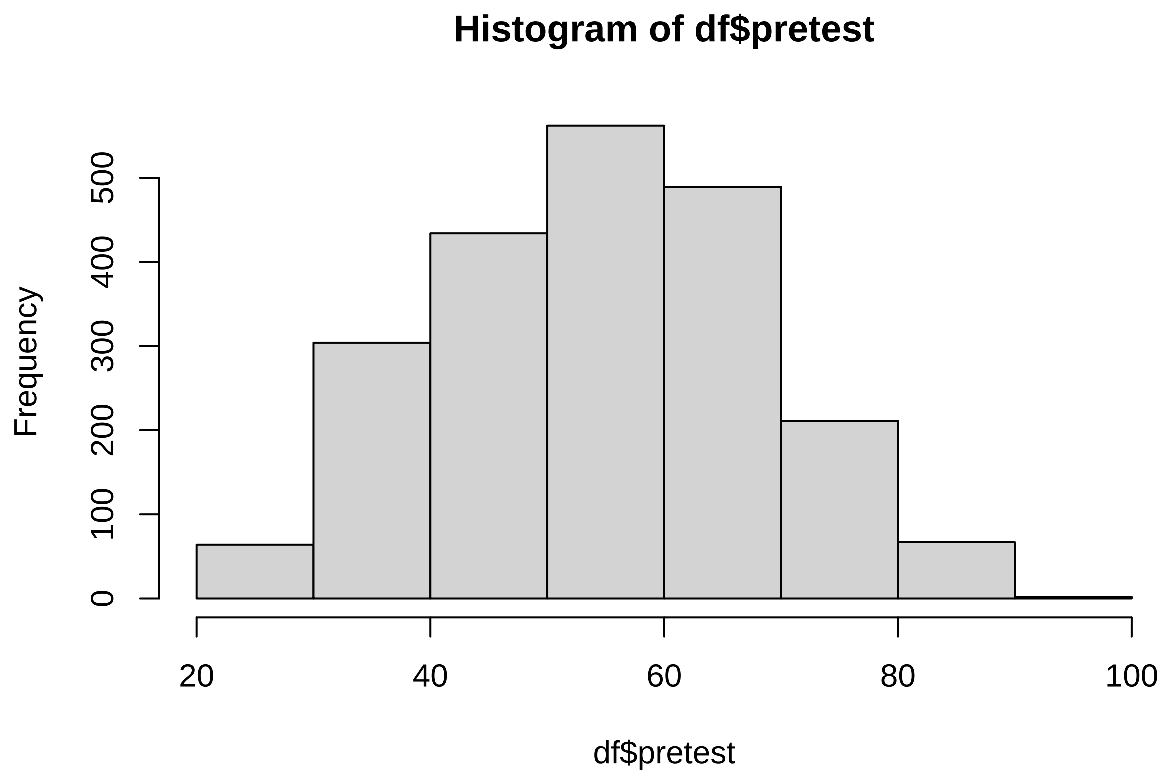


# Creating a Histogram

```
hist(df$pretest)
```



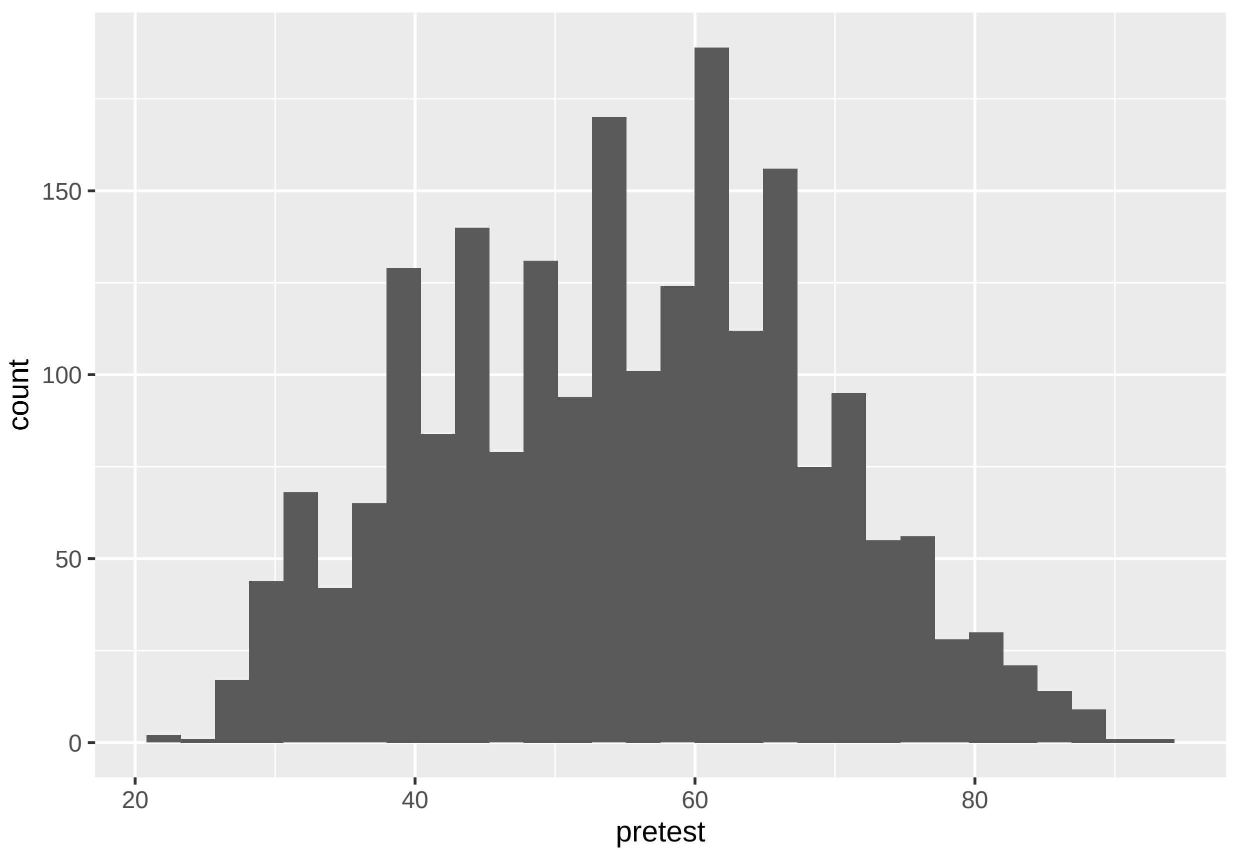
```r
# Specify approximate number of bins with breaks
hist(df$pretest, breaks = 10)
```



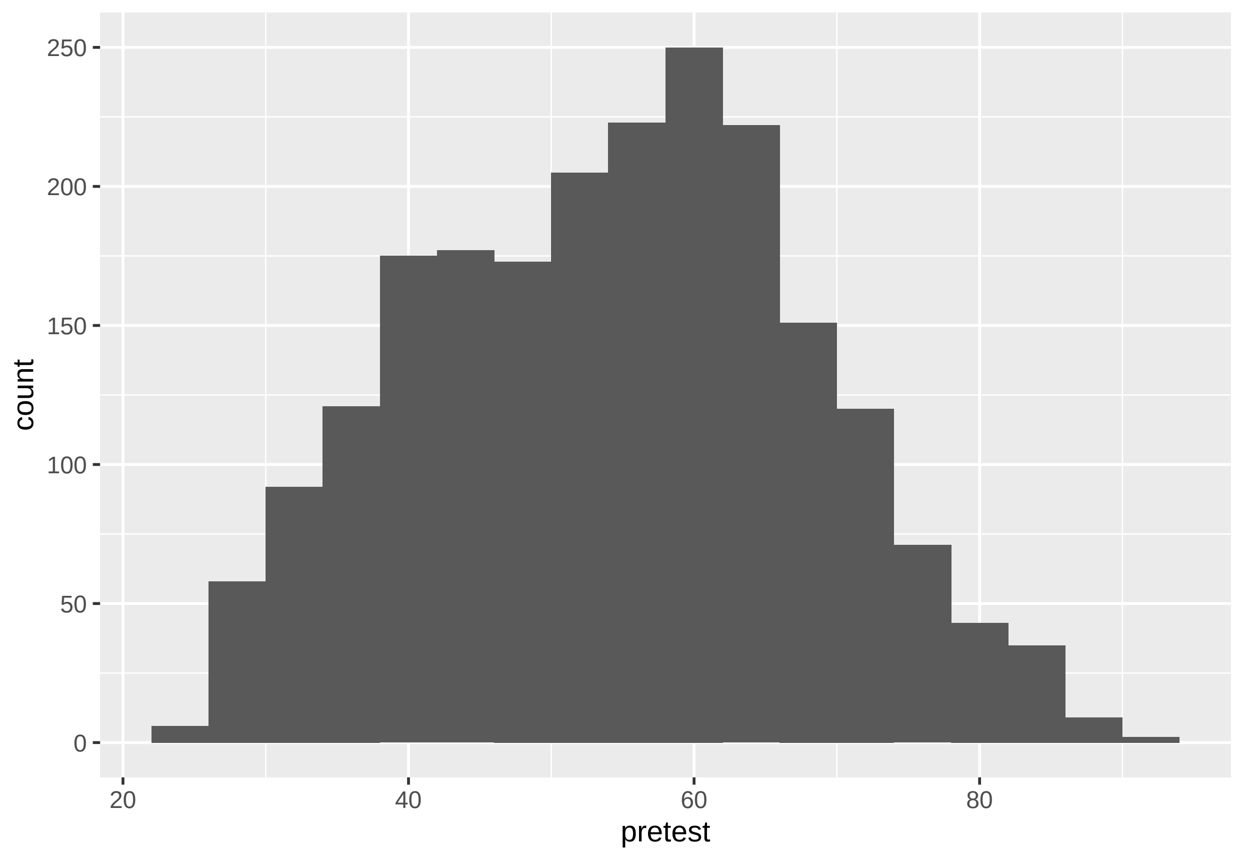
#ggplot2 histogram with default bin width (left); With wider bins (right)

```
ggplot(df, aes(x = pretest)) +
geom_histogram()
```

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

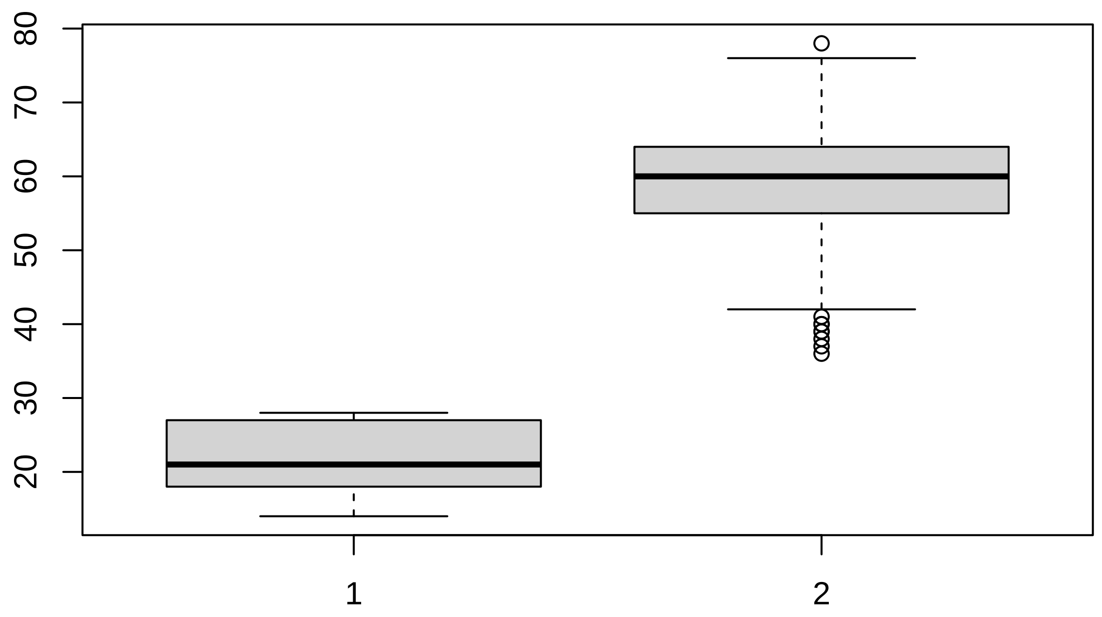


```r
ggplot(df, aes(x = pretest)) +
geom_histogram(binwidth = 4)
```



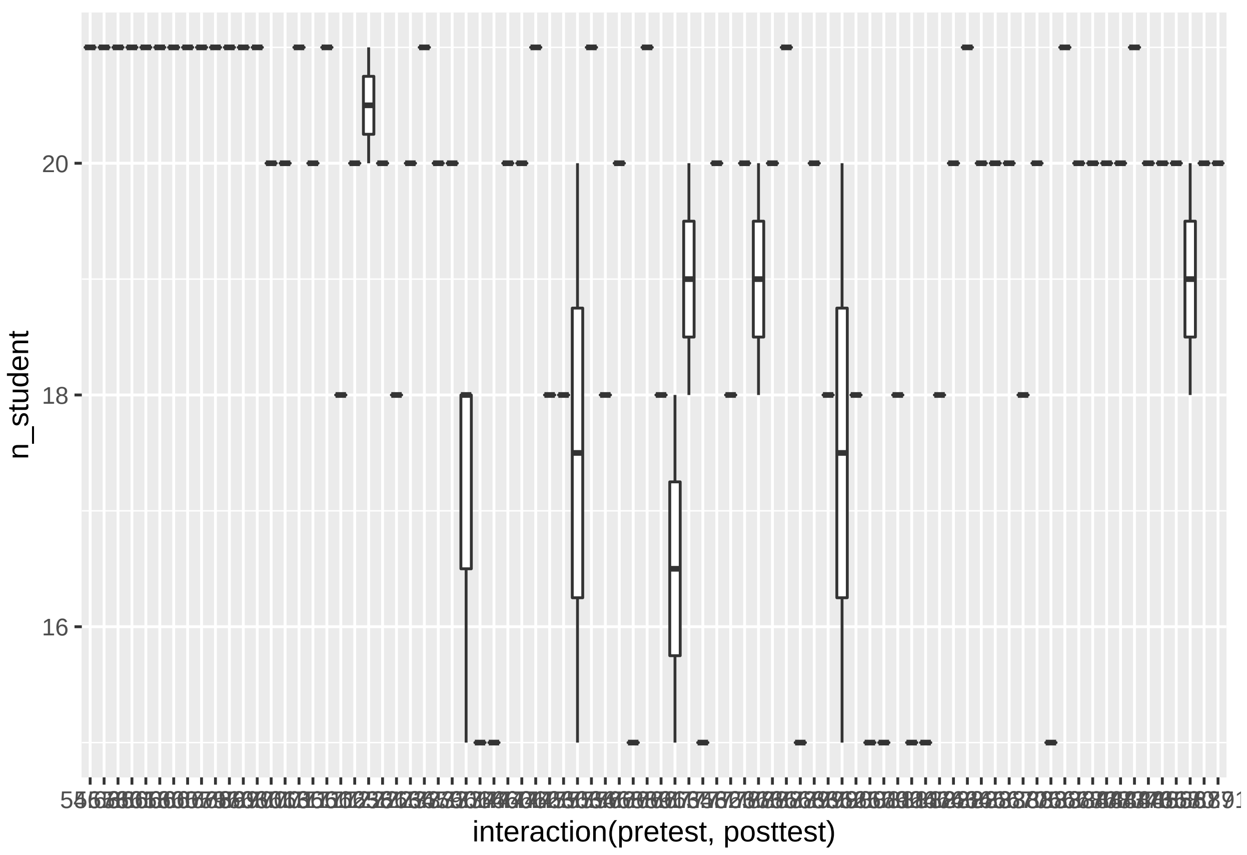
# Creating a Box Plot

```
boxplot(head(df$n_student,n=500), head(df$pretest,n=500))
```



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

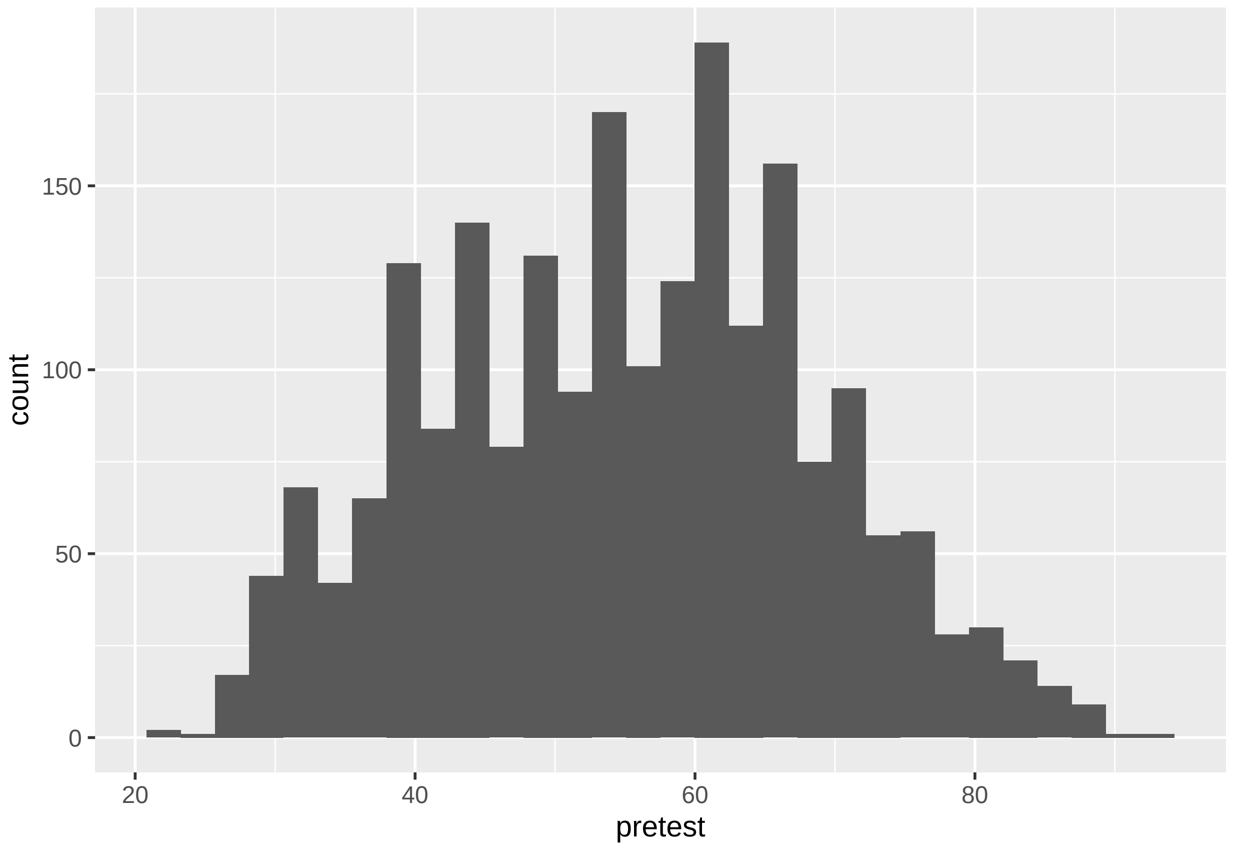
```
ggplot(head(df,n=100), aes(x = interaction(pretest, posttest), y = n_student)) +
geom_boxplot()
```



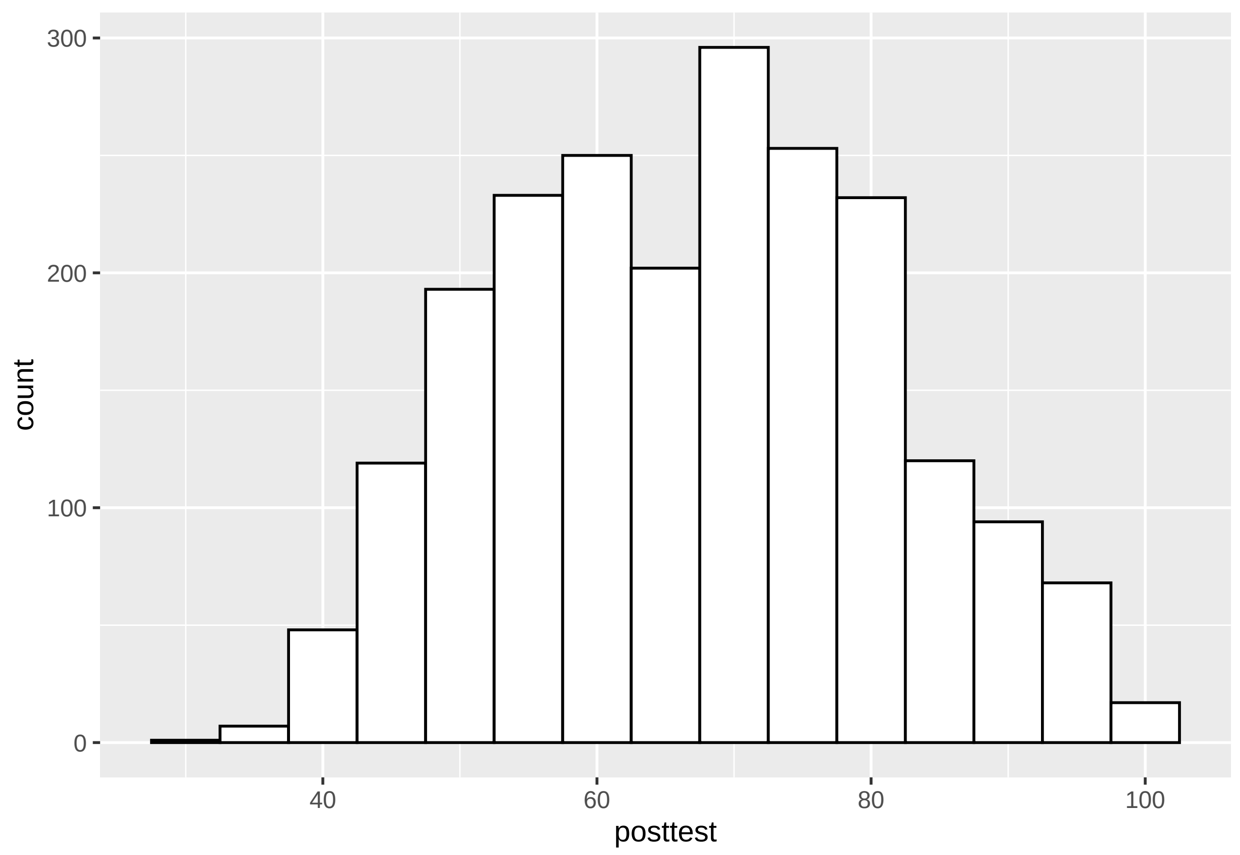
# Making a Basic Histogram

```r
ggplot(df, aes(x = pretest)) +geom_histogram()
```

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



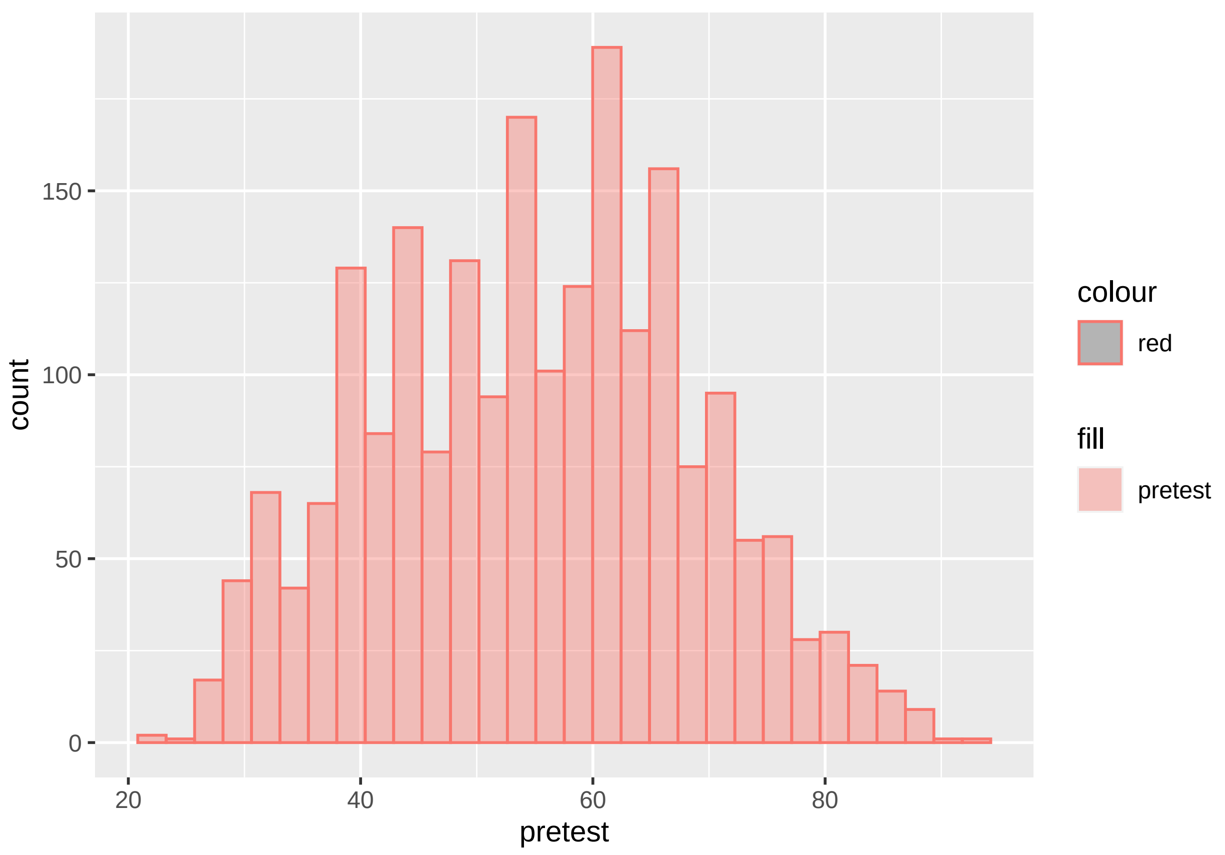
```
ggplot(df, aes(x= posttest)) +
geom_histogram(binwidth = 5, fill = "white", colour = "black")
```



# Map pretest to fill, make the bars NOT stacked, and make them semitransparent

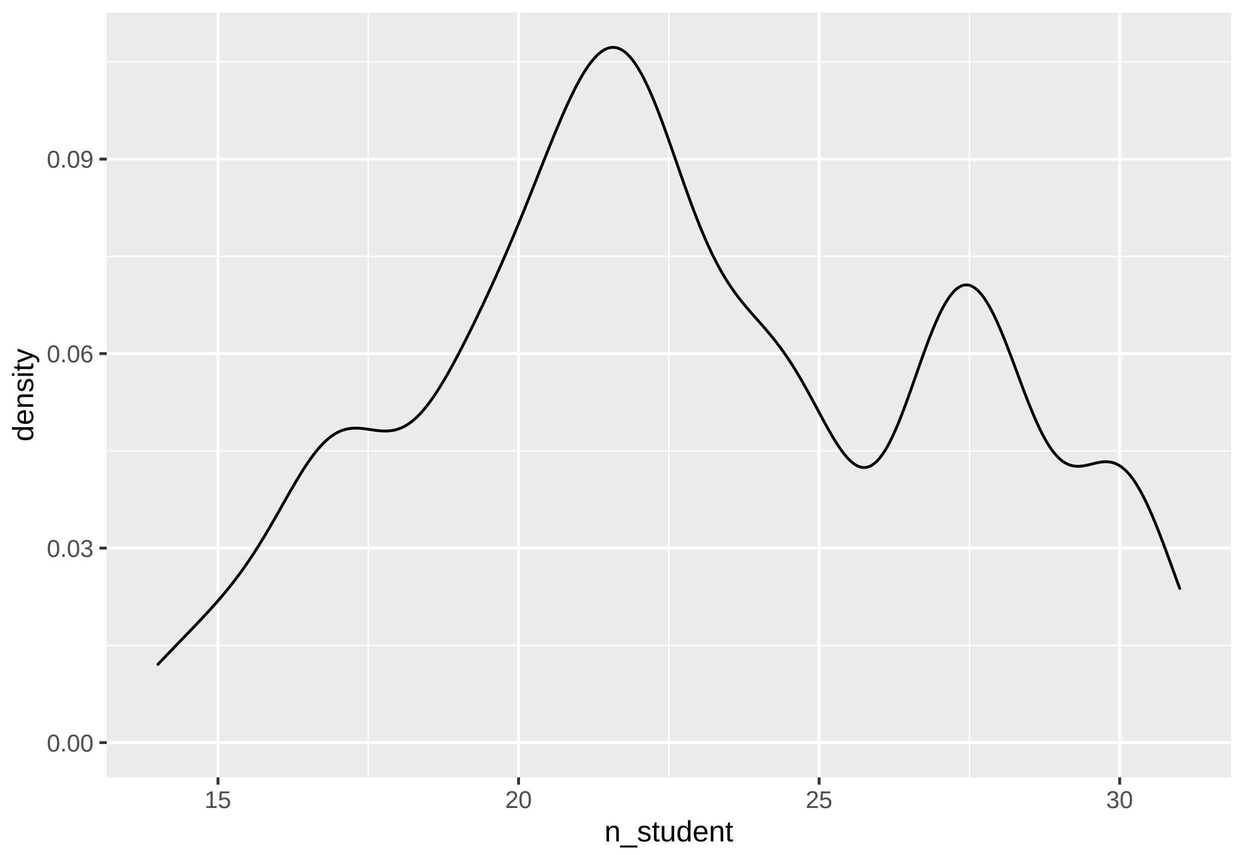
```
ggplot(df, aes(x = pretest, fill ='pretest',colour='red' )) +
geom_histogram(position = "identity", alpha = 0.4)
```

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

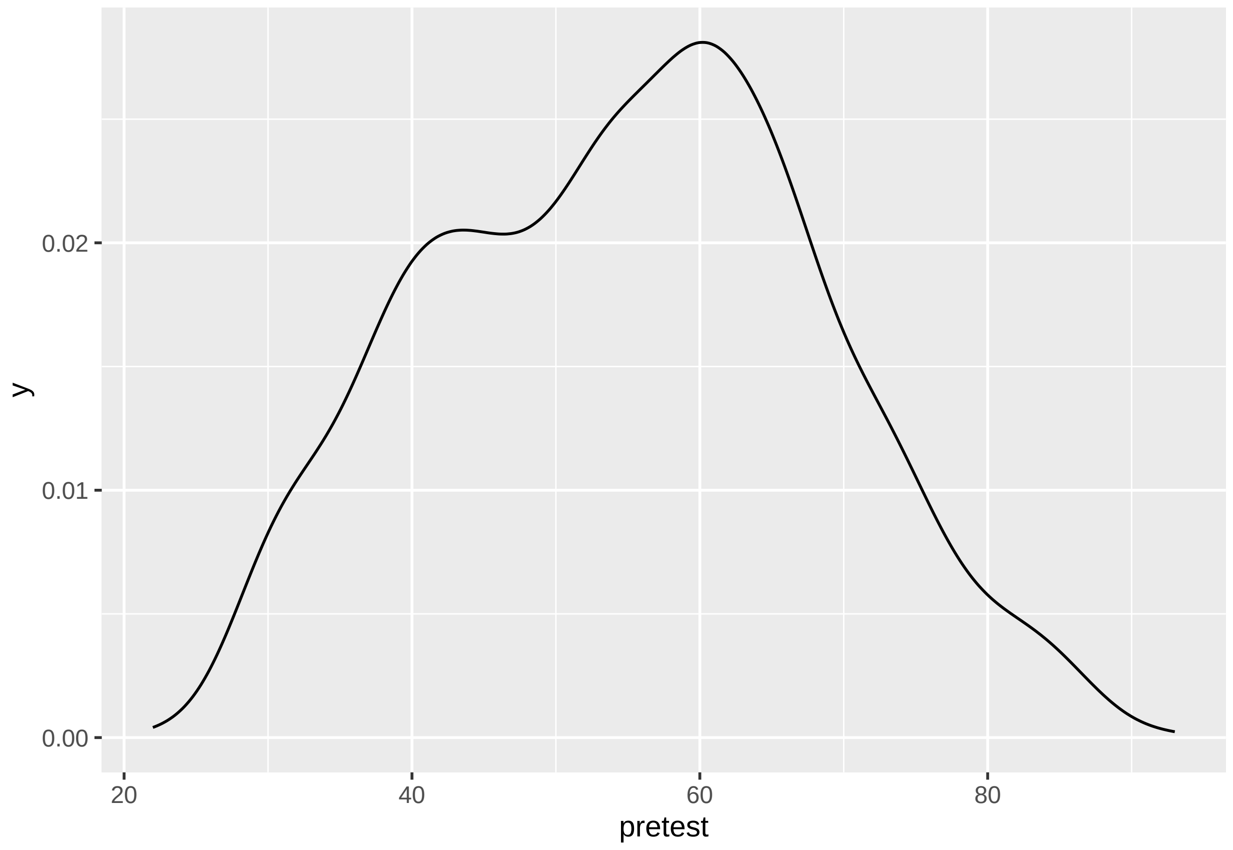


# Making a Density Curve

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

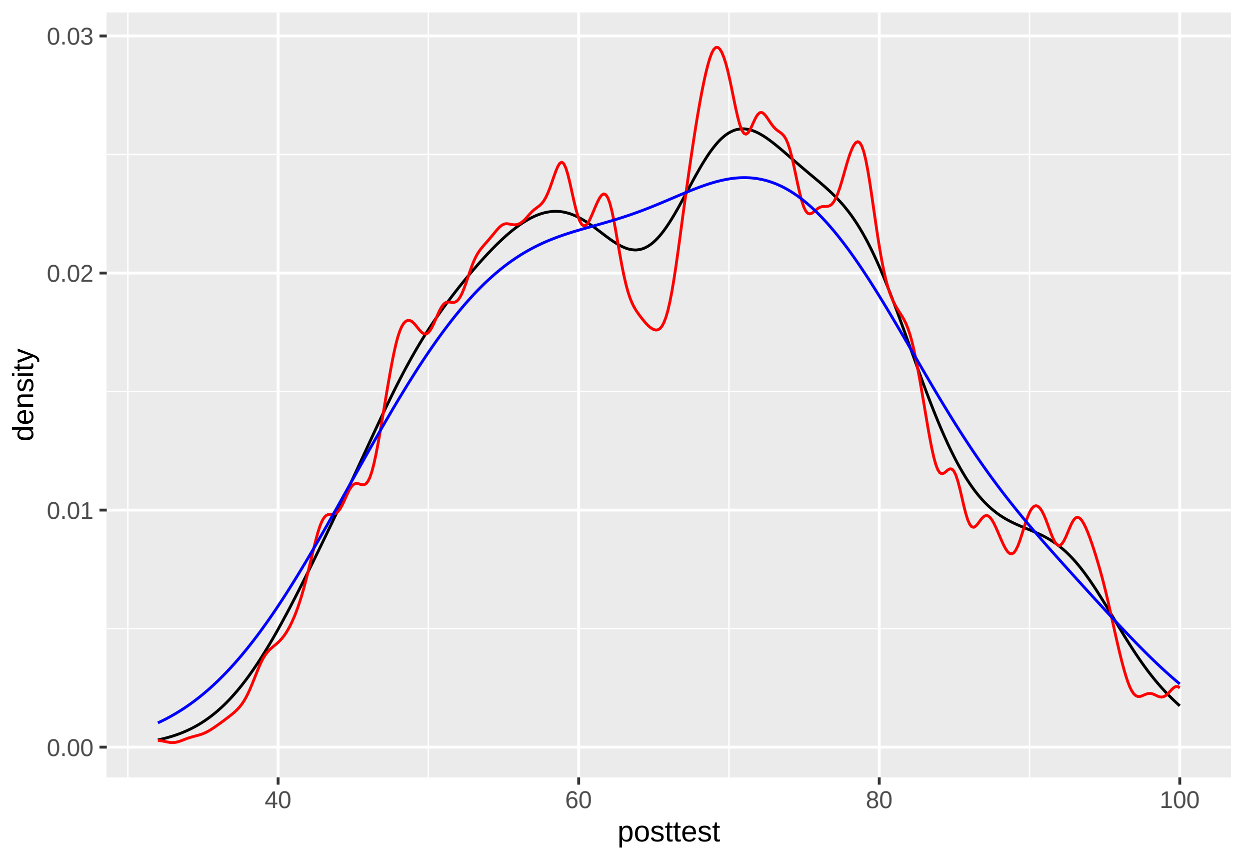


```
# expand_limits() increases the y range to include the value 0
ggplot(df, aes(x = pretest)) +geom_line(stat = "density") +expand_limits(y = 0)
```



# Density curve with a smaller and larger value of adjust:

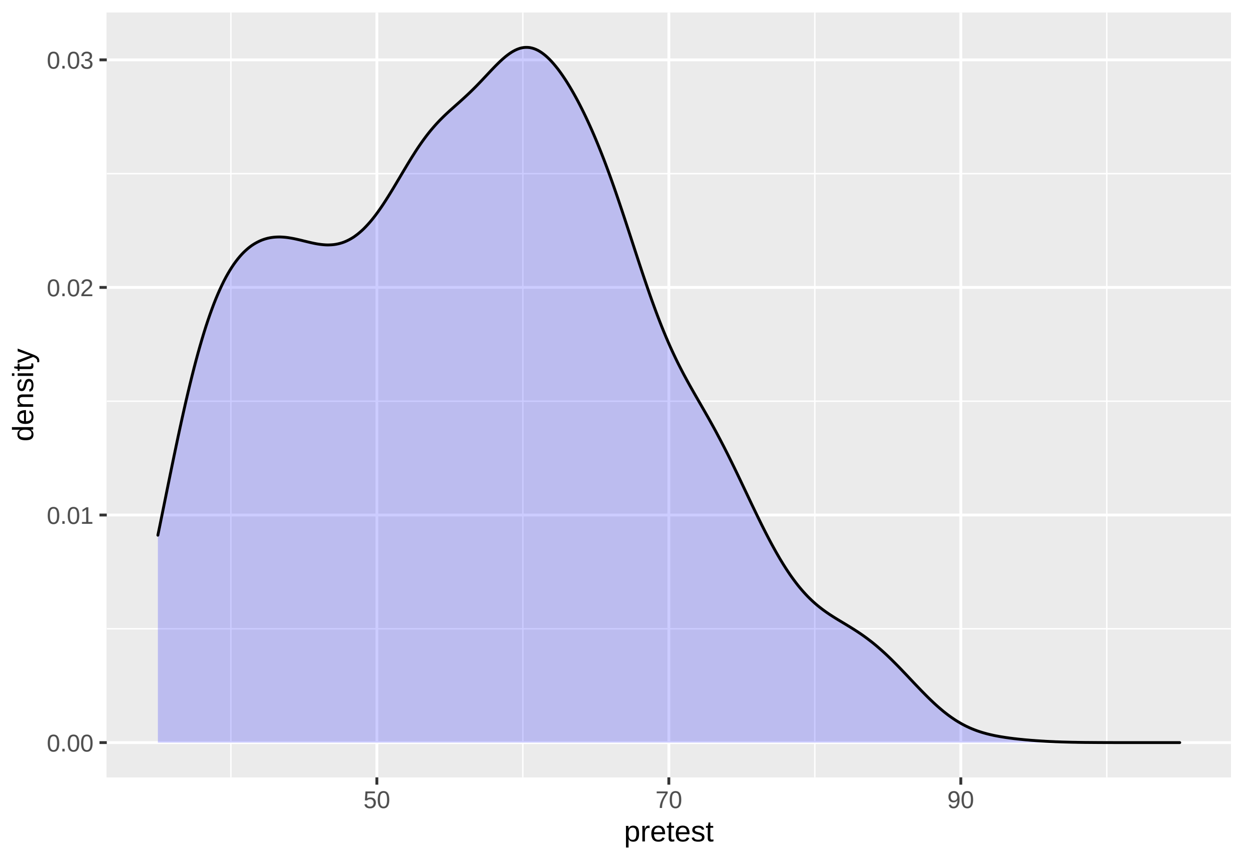
```
ggplot(df, aes(x = posttest)) +geom_line(stat = "density") +
geom_line(stat = "density", adjust = .25, colour = "red") +
geom_line(stat = "density", adjust = 2, colour = "blue")
```



# This draws a blue polygon with geom_density(), then adds a line on top # Density curve with wider x limits and a semitransparent fill (left); In two # parts, with geom_density() and geom_line() (right)

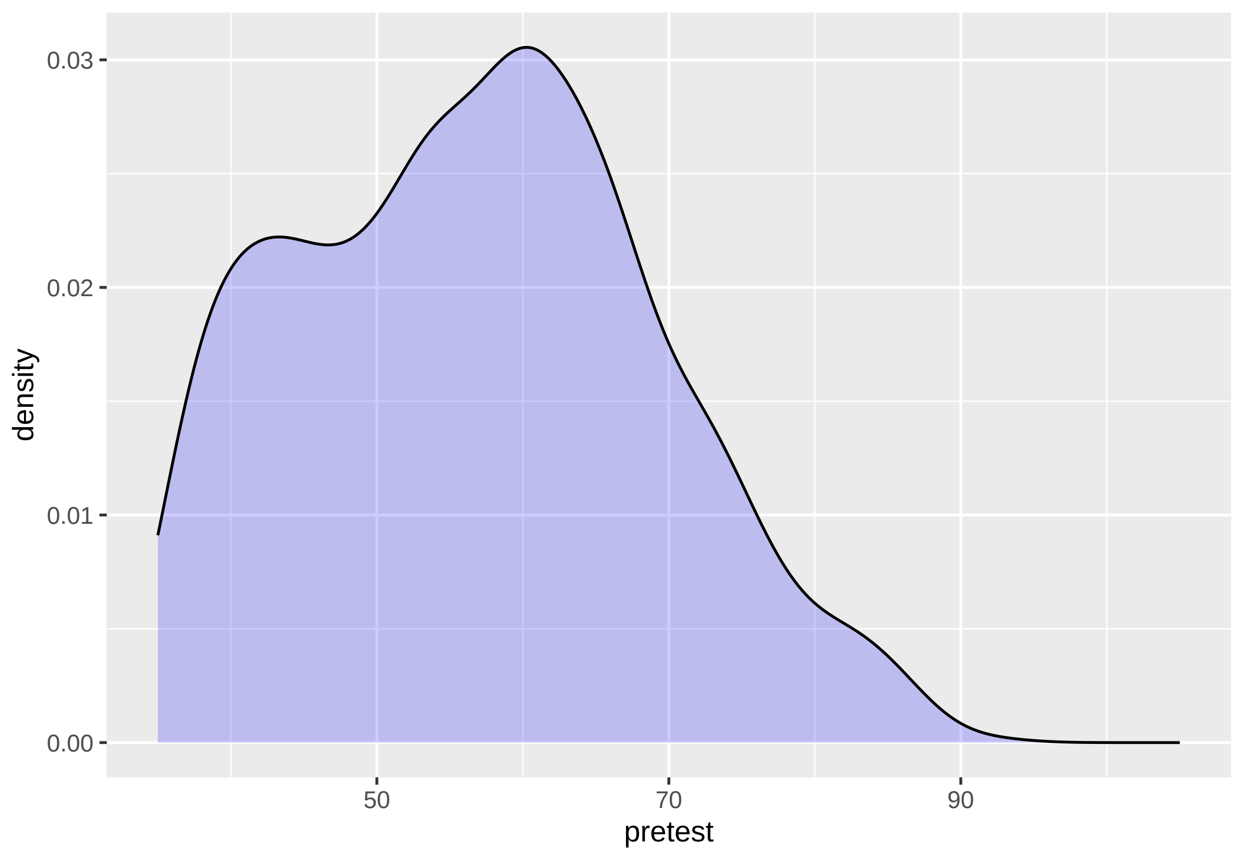
```
ggplot(df, aes(x = pretest)) +
geom_density(fill = "blue", alpha = .2) +
xlim(35, 105)
```

```
## Warning: Removed 156 rows containing non-finite values (stat_density).
```



```
ggplot(df, aes(x = pretest)) +
geom_density(fill = "blue", alpha = .2, colour = NA) +
xlim(35, 105) +
geom_line(stat = "density")
```

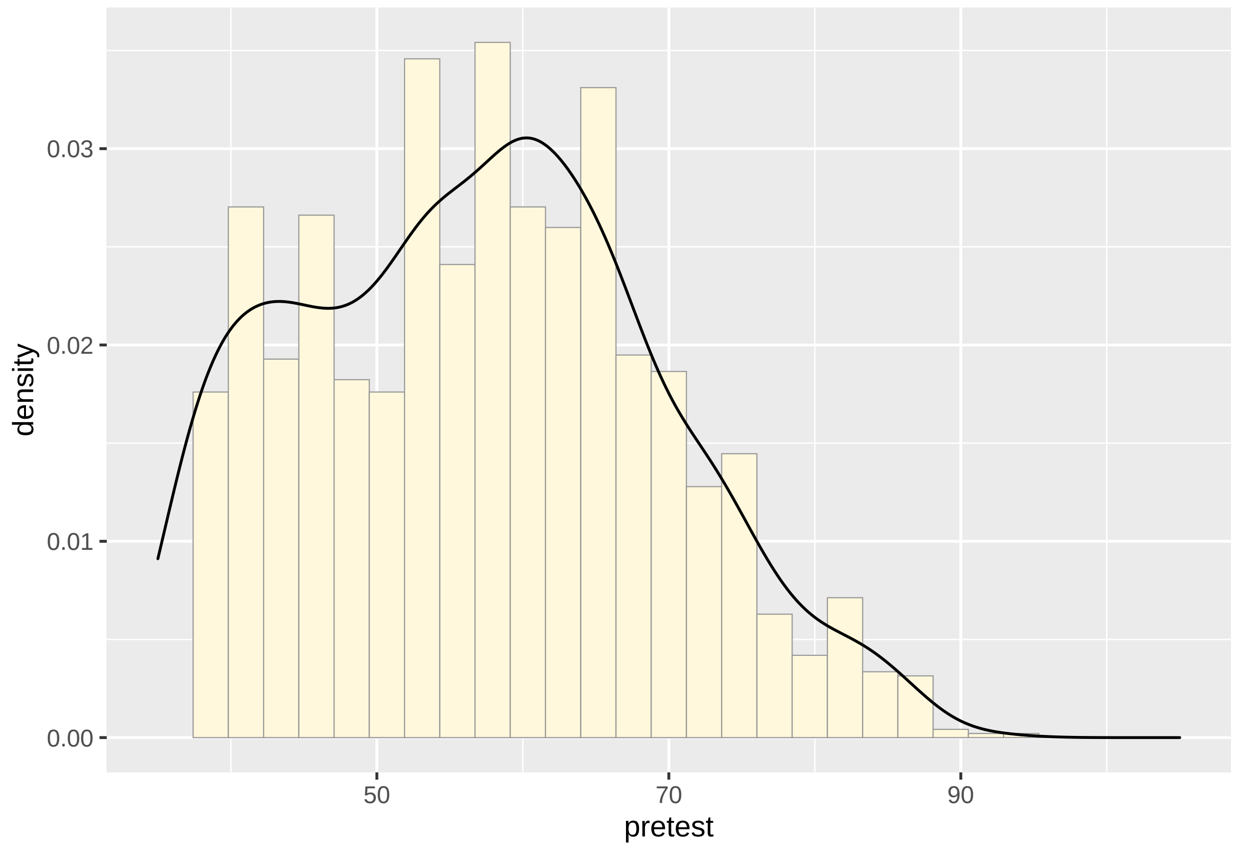
```
## Warning: Removed 156 rows containing non-finite values (stat_density).
## Removed 156 rows containing non-finite values (stat_density).
```



# Density curve overlaid on a histogram

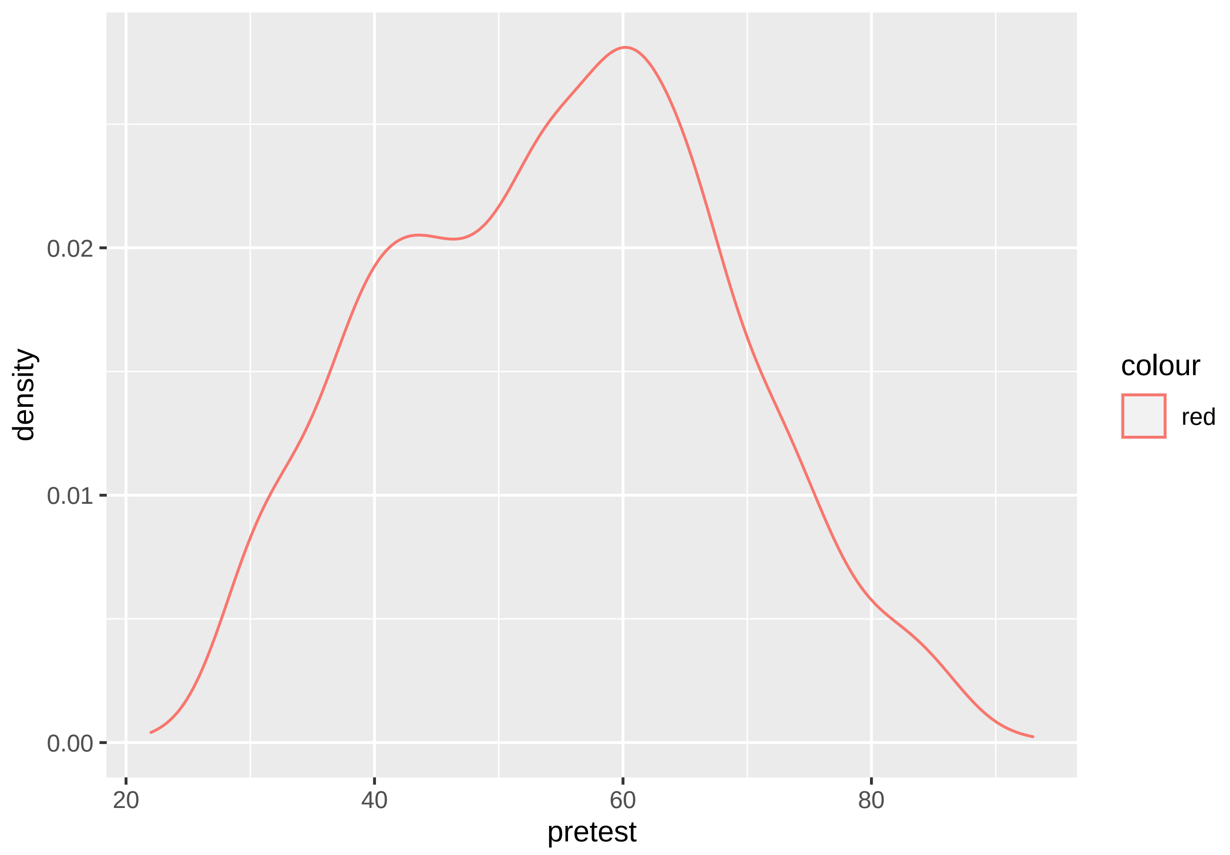
```
ggplot(df, aes(x = pretest, y = ..density..)) +geom_histogram(fill = "cornsilk", colour = "grey60", size
```

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

## Warning: Removed 156 rows containing non-finite values (stat_bin).

## Warning: Removed 156 rows containing non-finite values (stat_density).

## Warning: Removed 1 rows containing missing values (geom_bar).
```

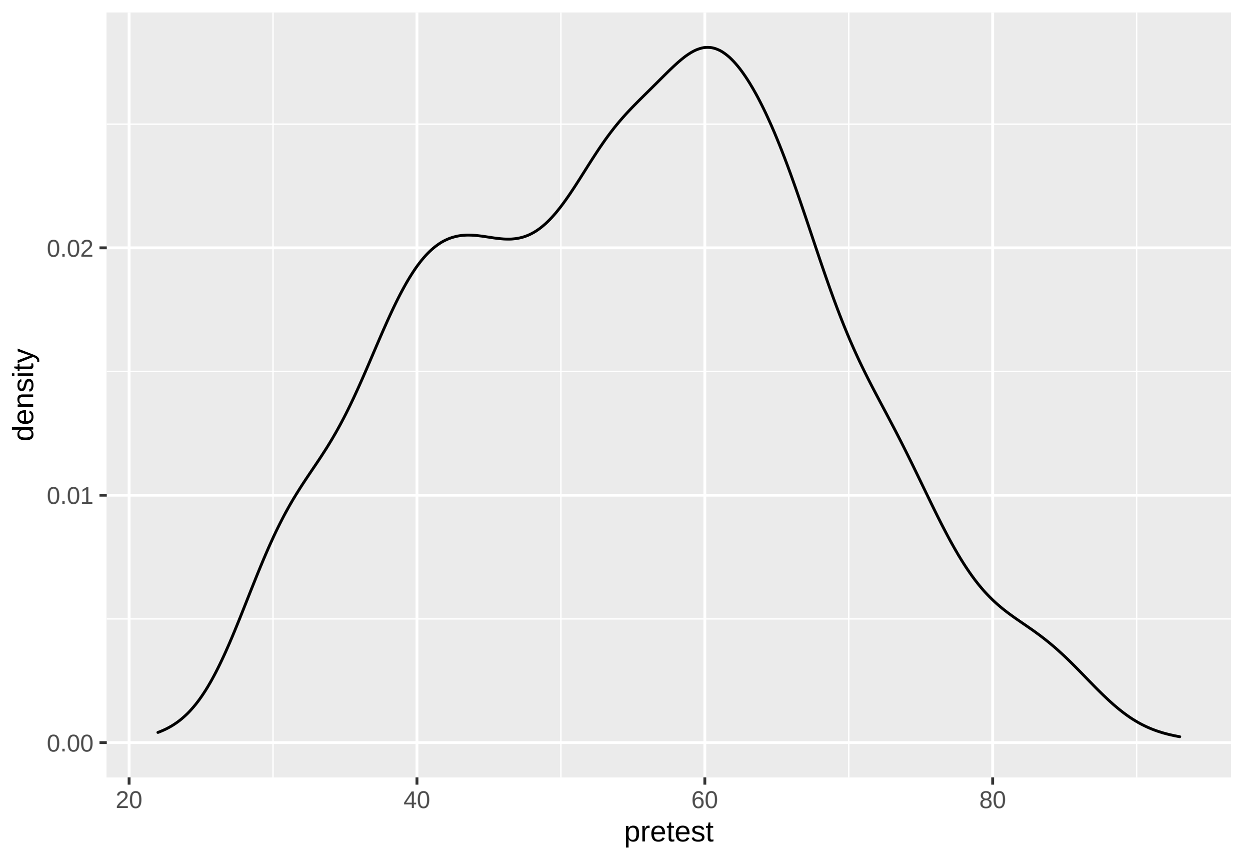


# Making Multiple Density Curves from Grouped Data

```
data5 <- df %>%
mutate(n_student = as.factor(n_student)) # Convert n_student to a factor
# Map n_student to colour
ggplot(data5, aes(x = pretest, colour = "red")) +geom_density()
```



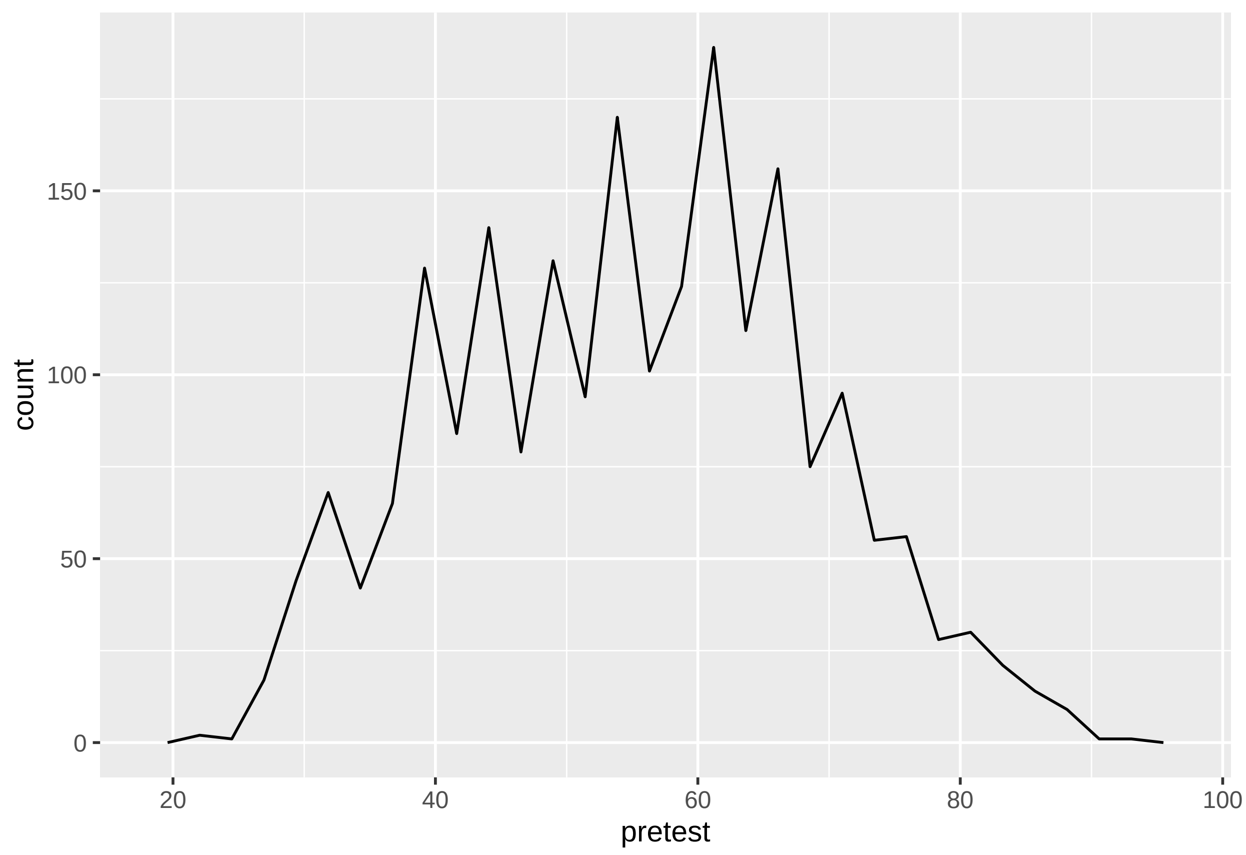
```
# Map n_student to fill and make the fill semitransparent by setting alpha
ggplot(data5, aes(x = pretest, fill = pretest)) +geom_density(alpha = .3)
```



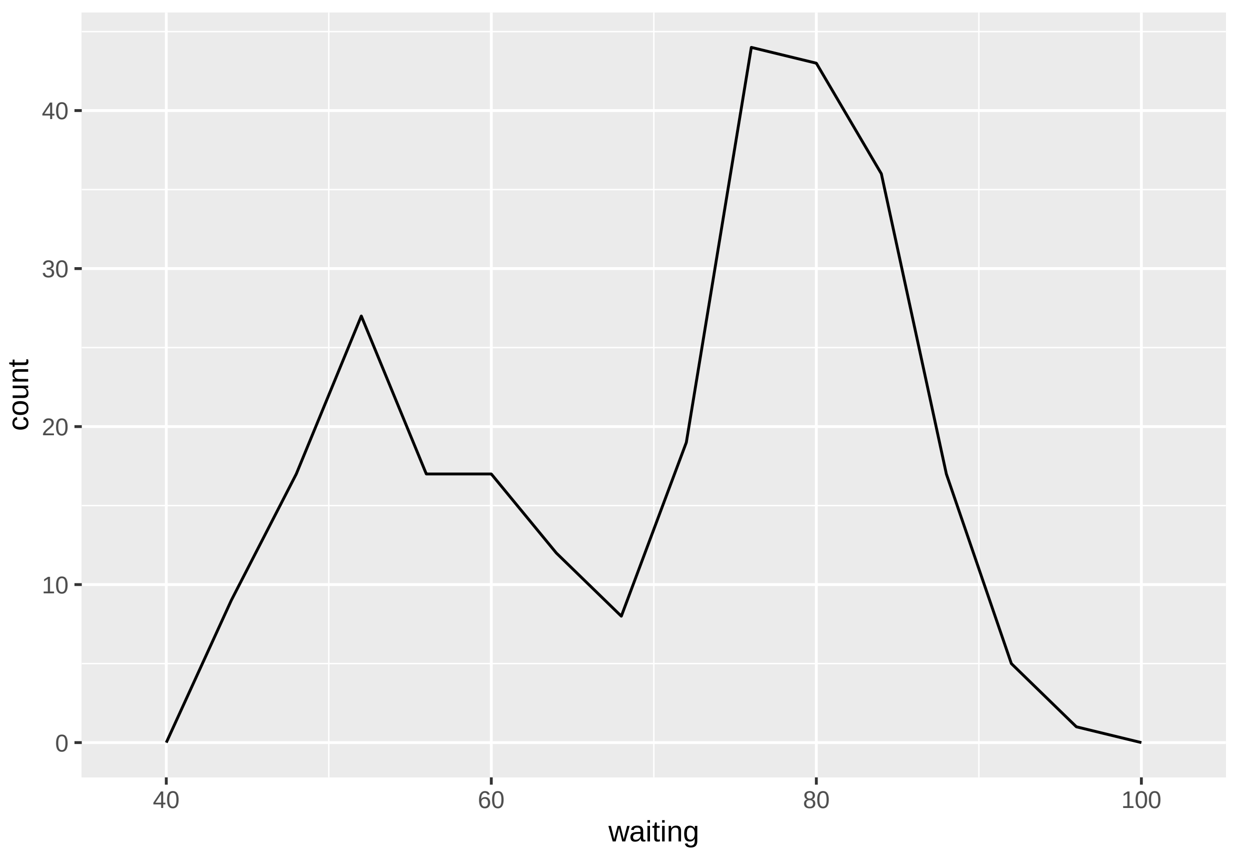
# Making a Frequency Polygon

```
ggplot(df, aes(x=pretest)) +
geom_freqpoly()
```

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

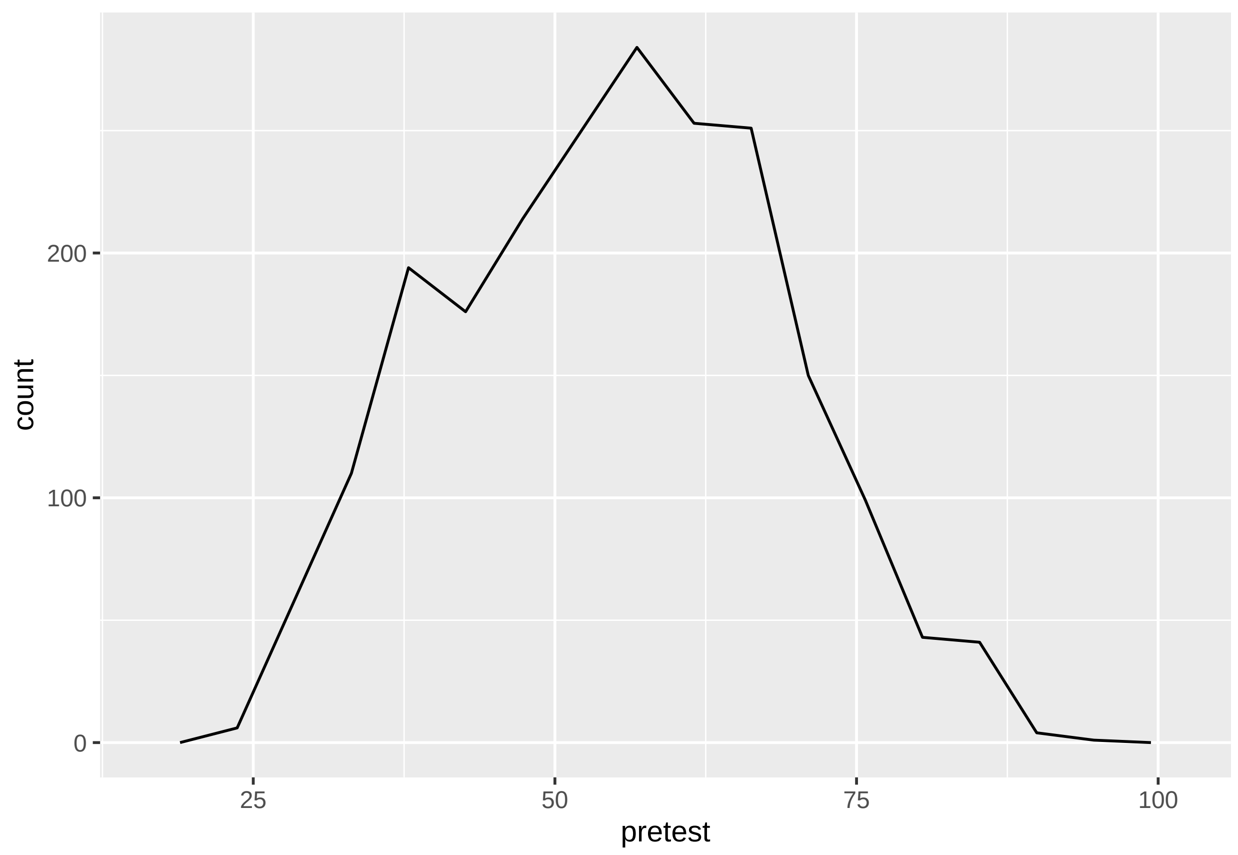


```
ggplot(faithful, aes(x = waiting)) +
geom_freqpoly(binwidth = 4)           #controlling bin width
```



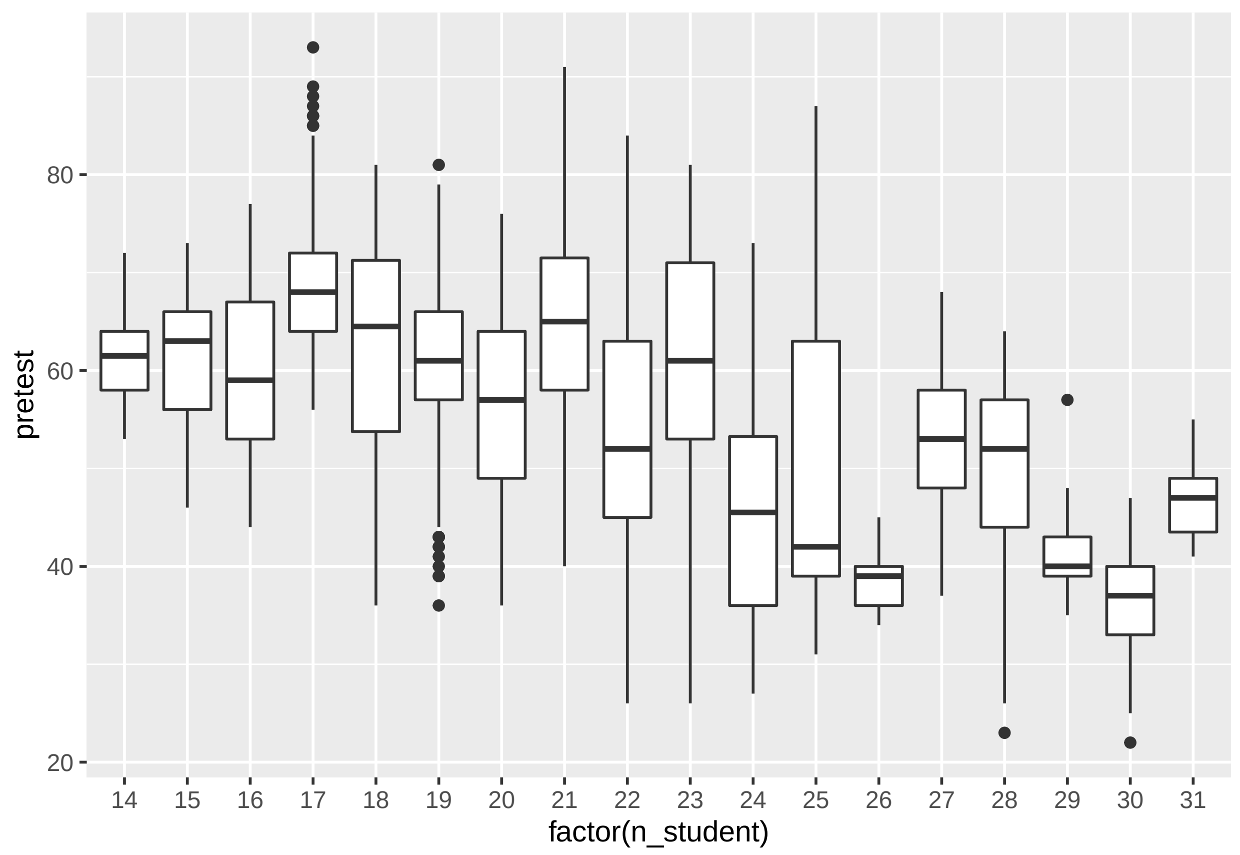
# Divide the x-axis range into 15 bins

```
binsize <- diff(range(df$pretest))/15
ggplot(df, aes(x = pretest)) +
geom_freqpoly(binwidth = binsize)
```



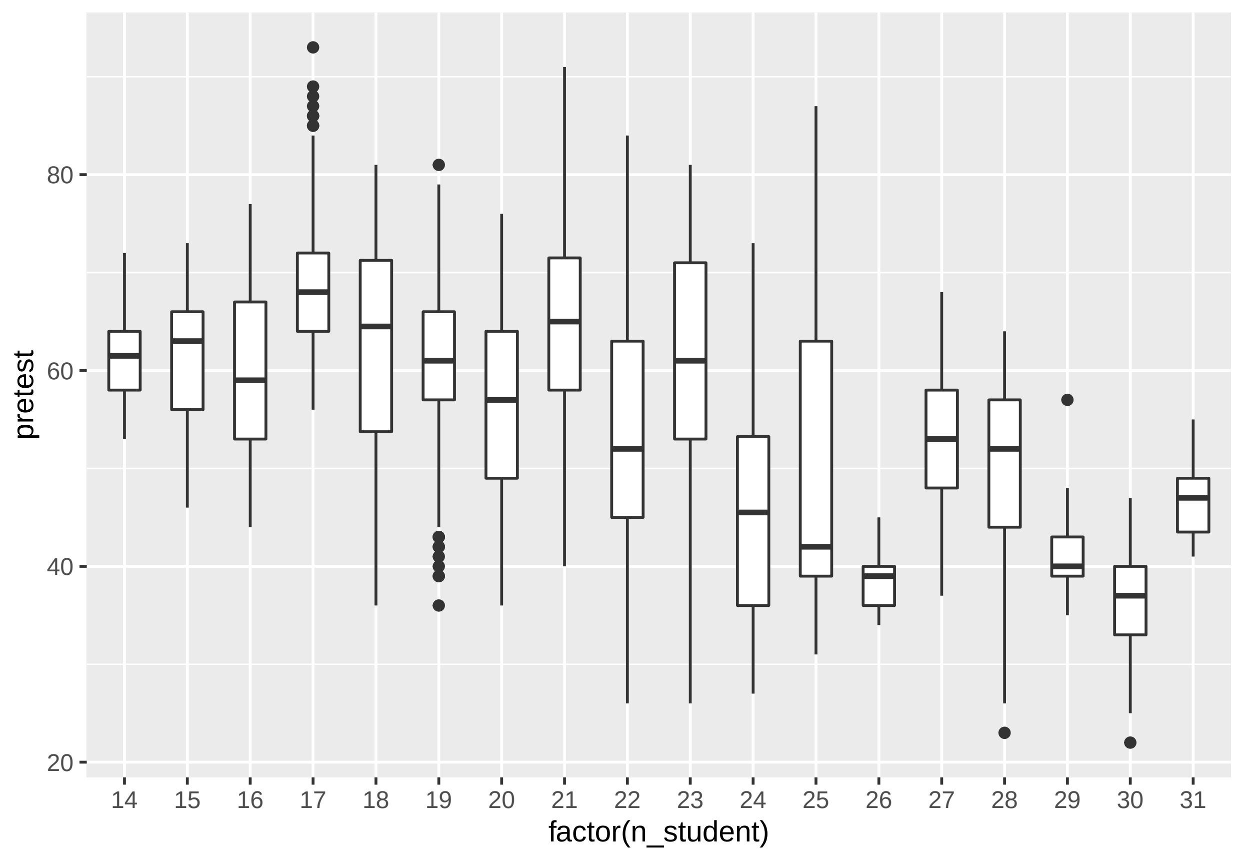
# Making a Basic Box Plot

```
ggplot(df, aes(x = factor(n_student), y = pretest)) +
geom_boxplot()
```

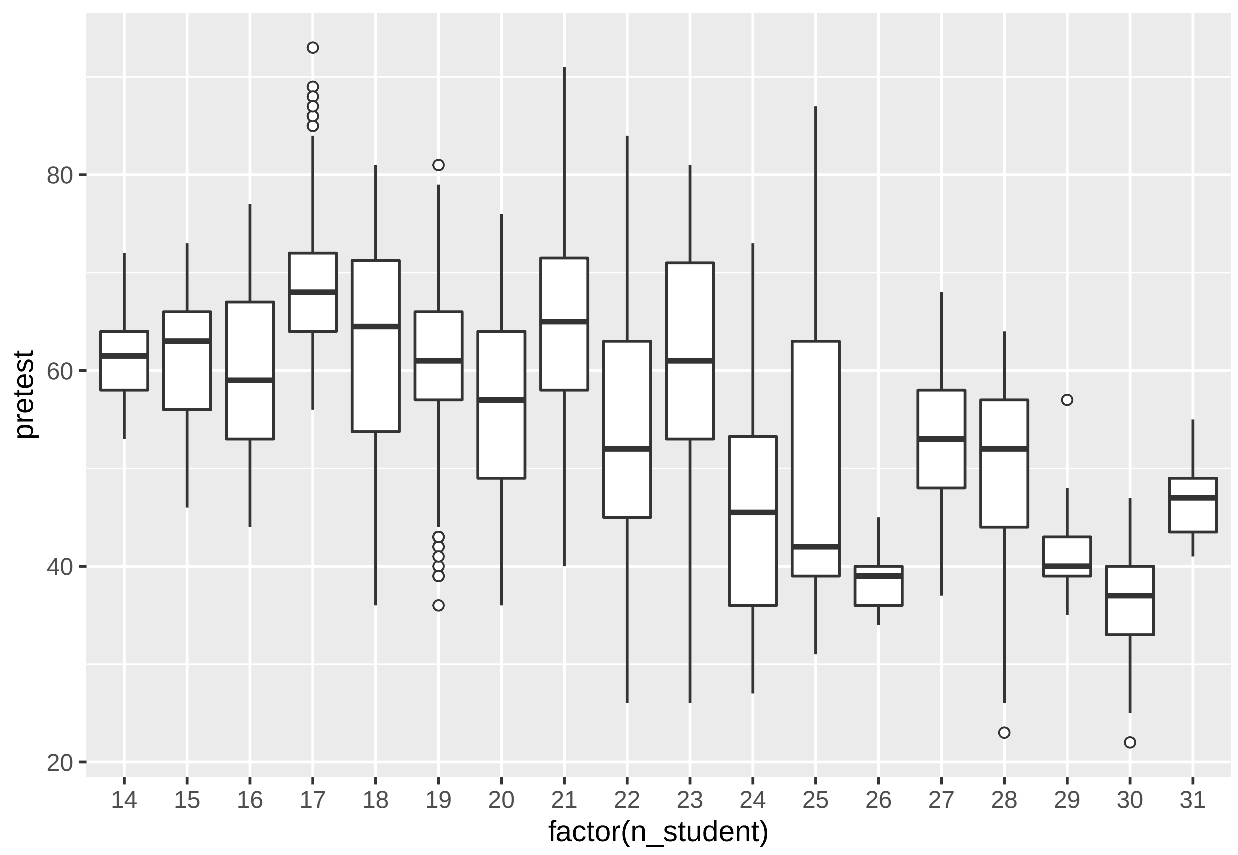


# Box plot with narrower boxes (1) # With smaller, hollow outlier points(2)

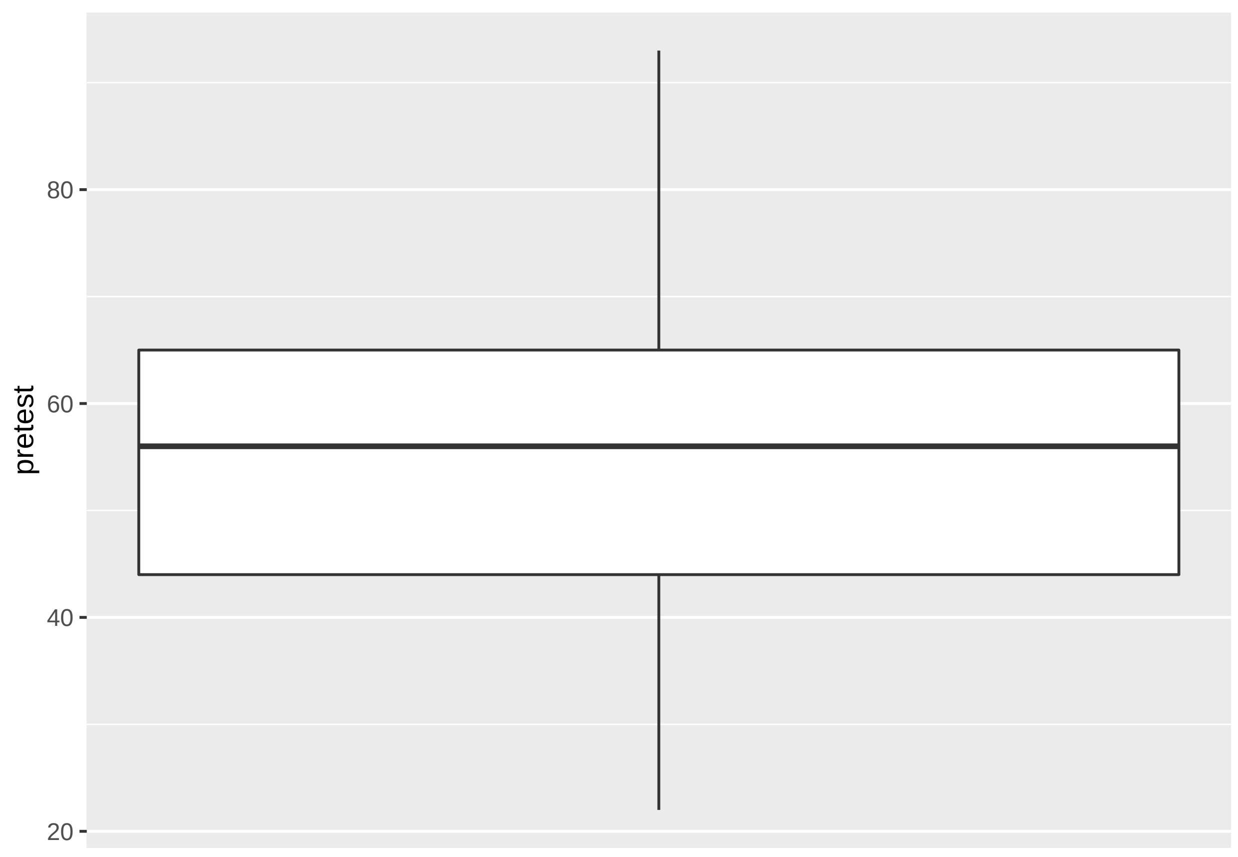
```
ggplot(df, aes(x = factor(n_student), y = pretest)) +geom_boxplot(width = .5)
```



```
ggplot(df, aes(x = factor(n_student), y = pretest)) +geom_boxplot(outlier.size = 1.5, outlier.shape = 2
```



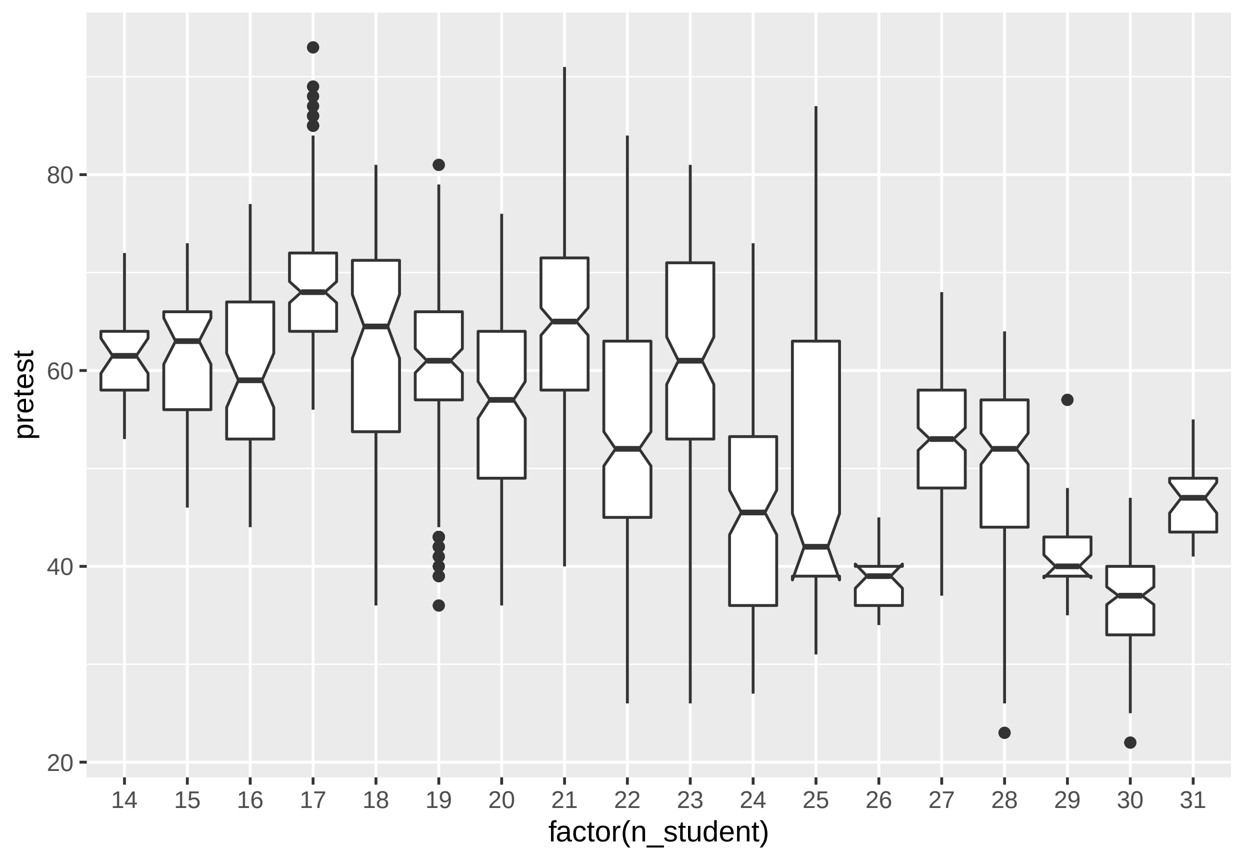
```
ggplot(df, aes(x = 1, y = pretest)) +geom_boxplot() +scale_x_continuous(breaks = NULL) +theme(axis.titl
```



Adding notches to a box plot to assess whether the medians are different.

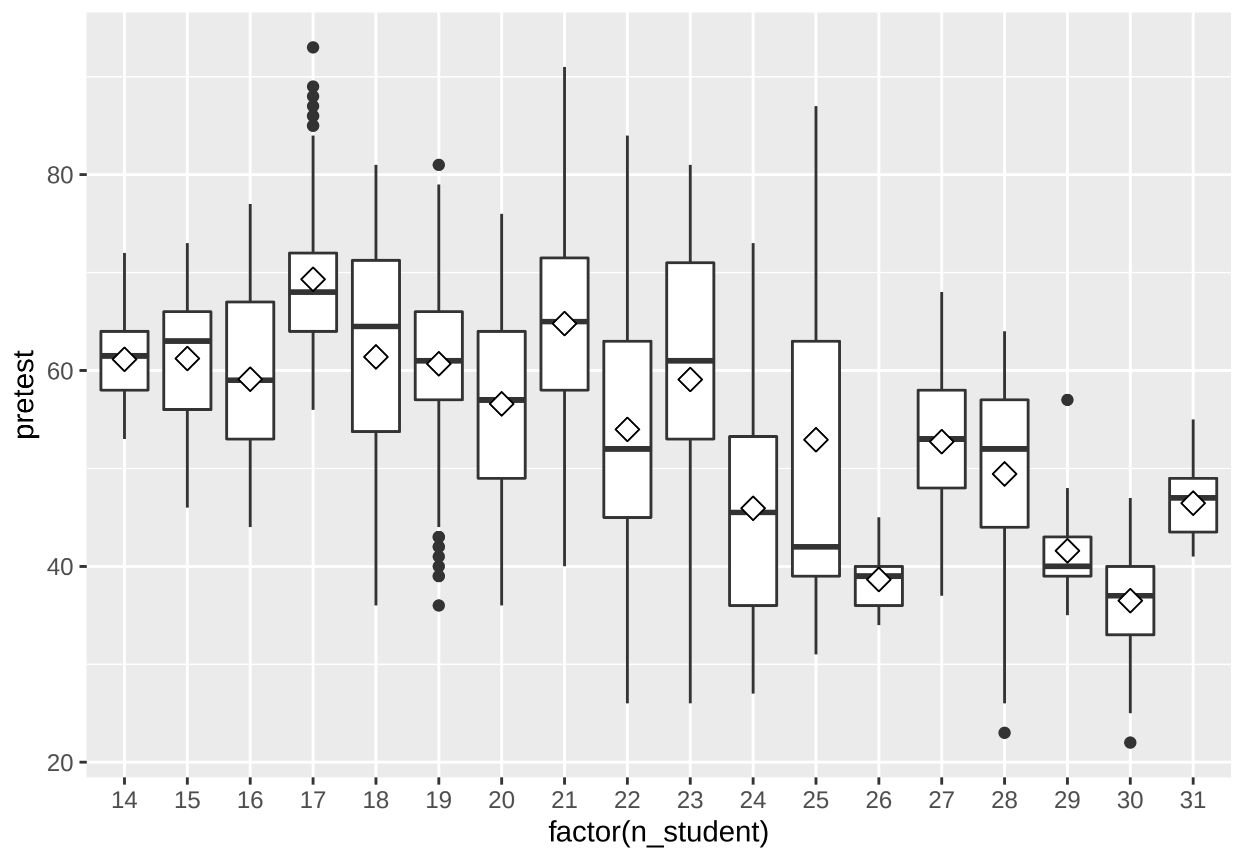
```
ggplot(df, aes(x = factor(n_student), y = pretest)) +geom_boxplot(notch = TRUE)
```

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



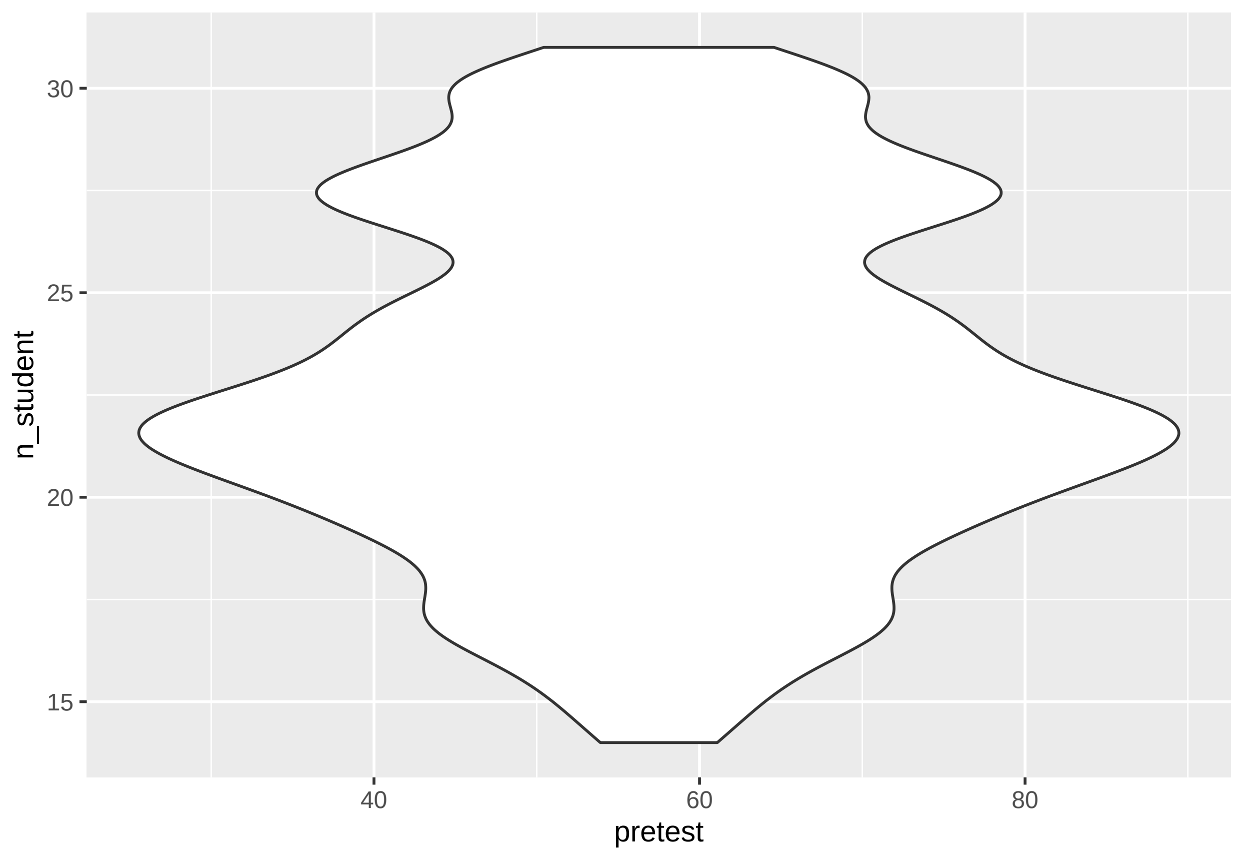
# Adding means to box plot

```
ggplot(df, aes(x = factor(n_student), y = pretest)) +geom_boxplot() +stat_summary(fun = "mean", geom =
fill = "white")
```



# Making a Violin Plot

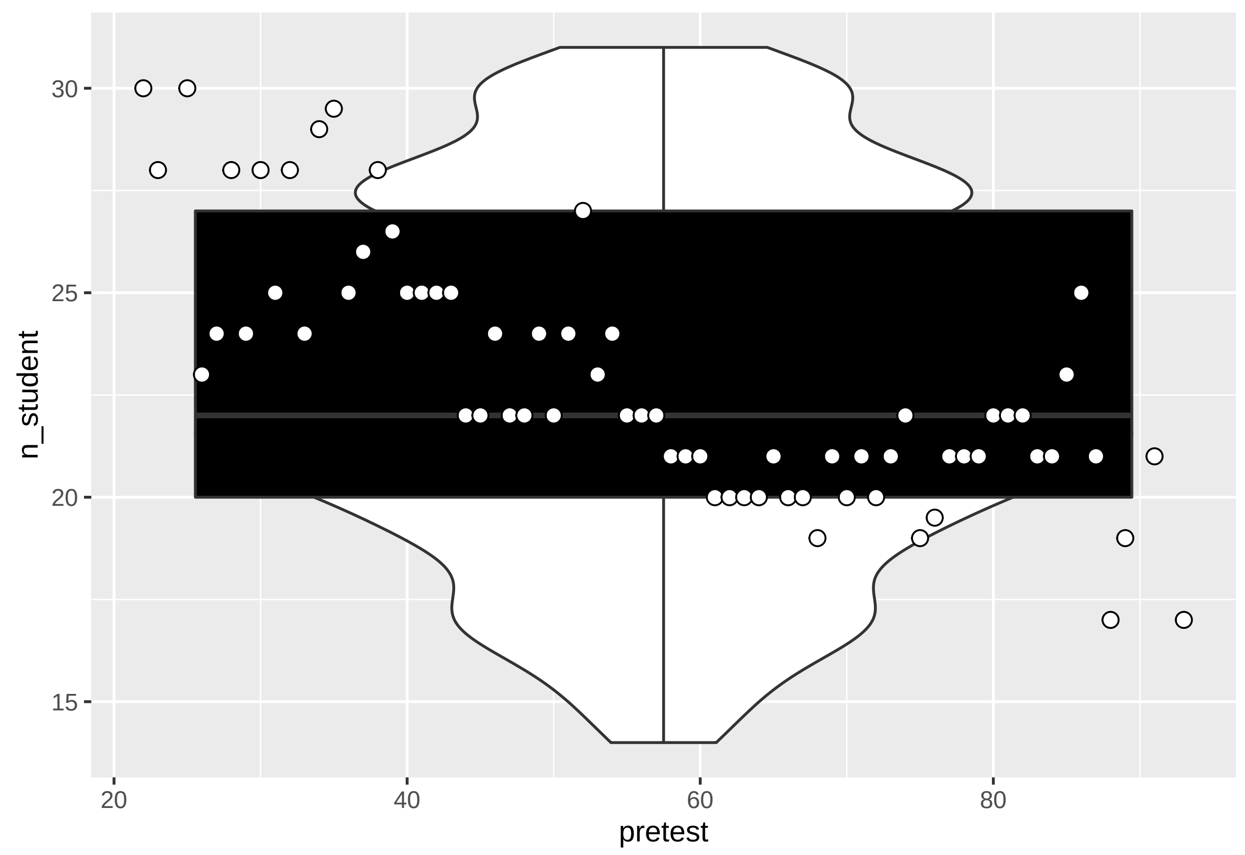
```
data6 <- ggplot(df, aes(x = pretest, y = n_student))
data6+geom_violin()
```



# A violin plot with box plot overlaid on it

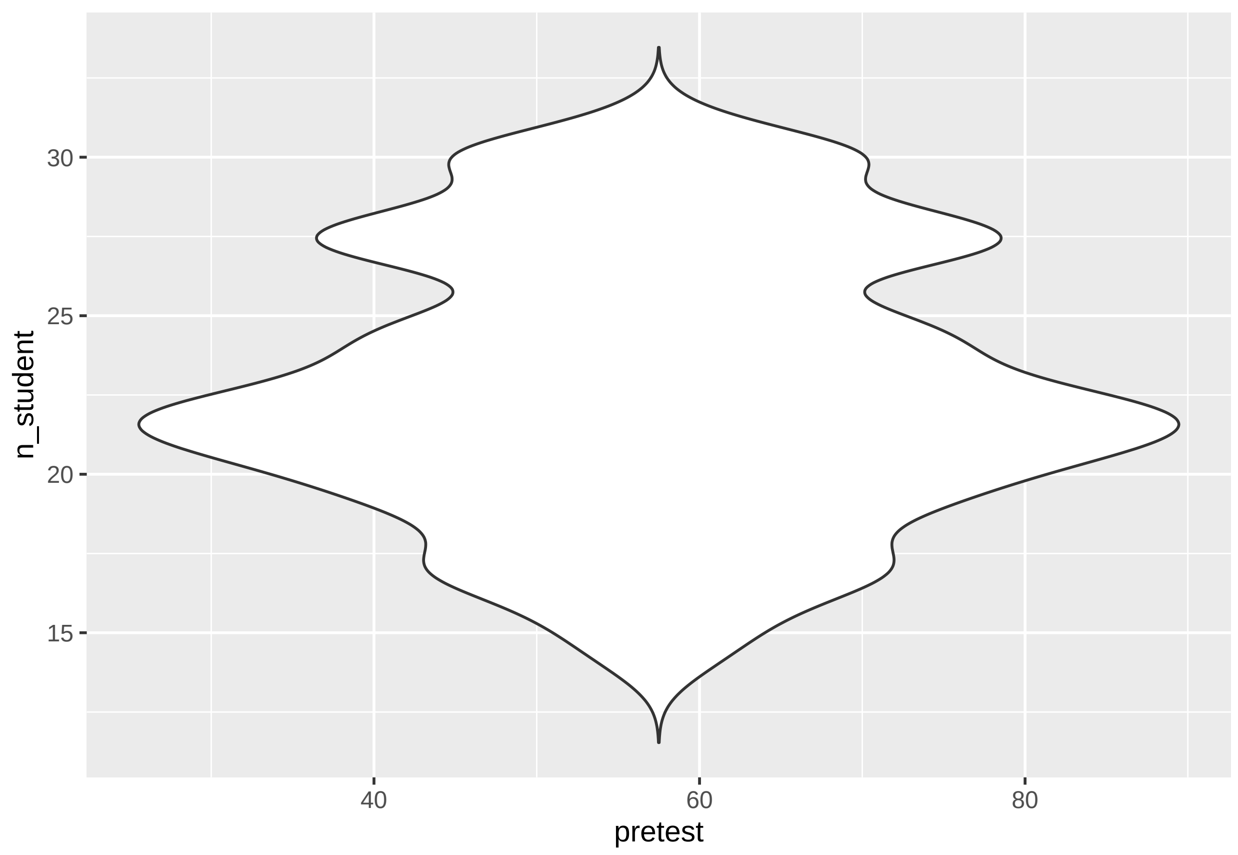
```
data6+geom_violin() +geom_boxplot(width = .1, fill = "black", outlier.colour = NA) +
stat_summary(fun= median, geom = "point", fill = "white", shape = 21,
size = 2.5)
```

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



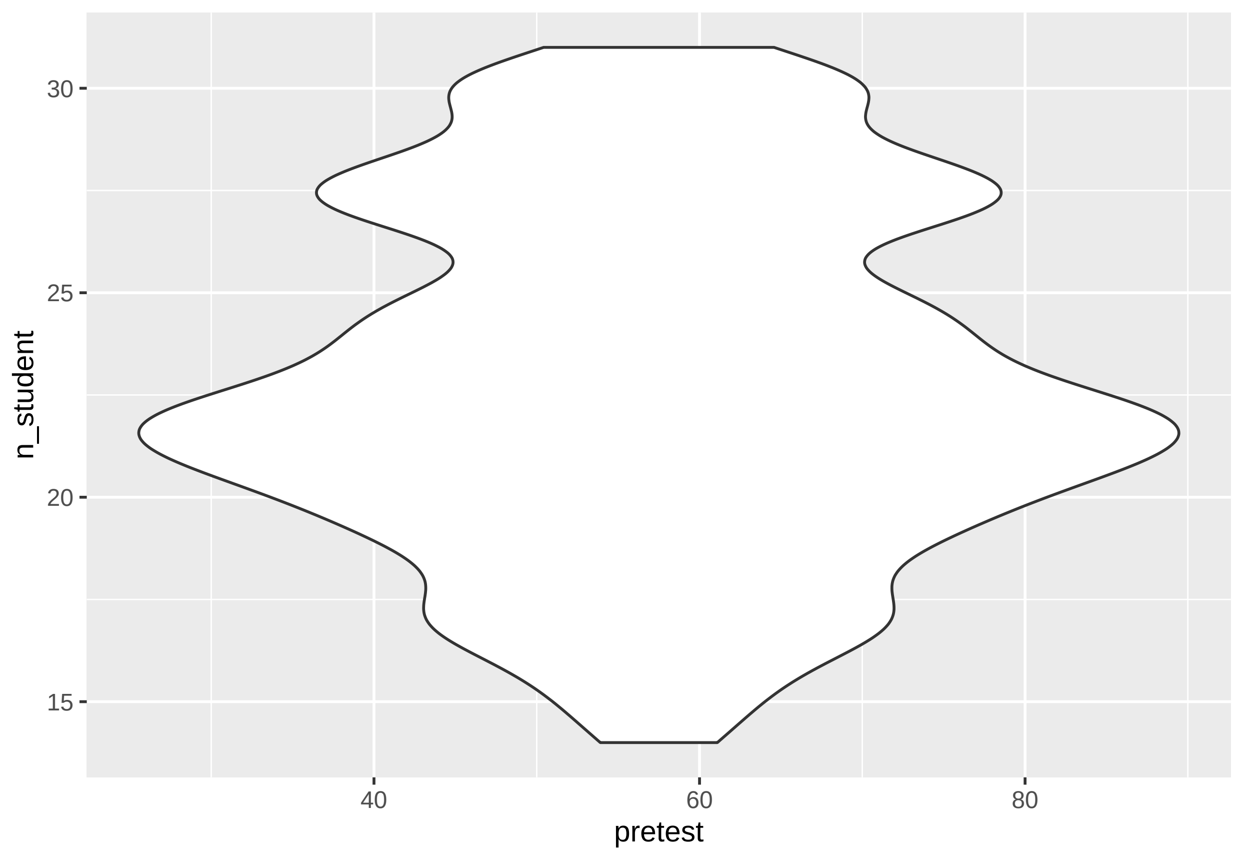
# A violin plot with tails

```
data6+geom_violin(trim = FALSE)
```



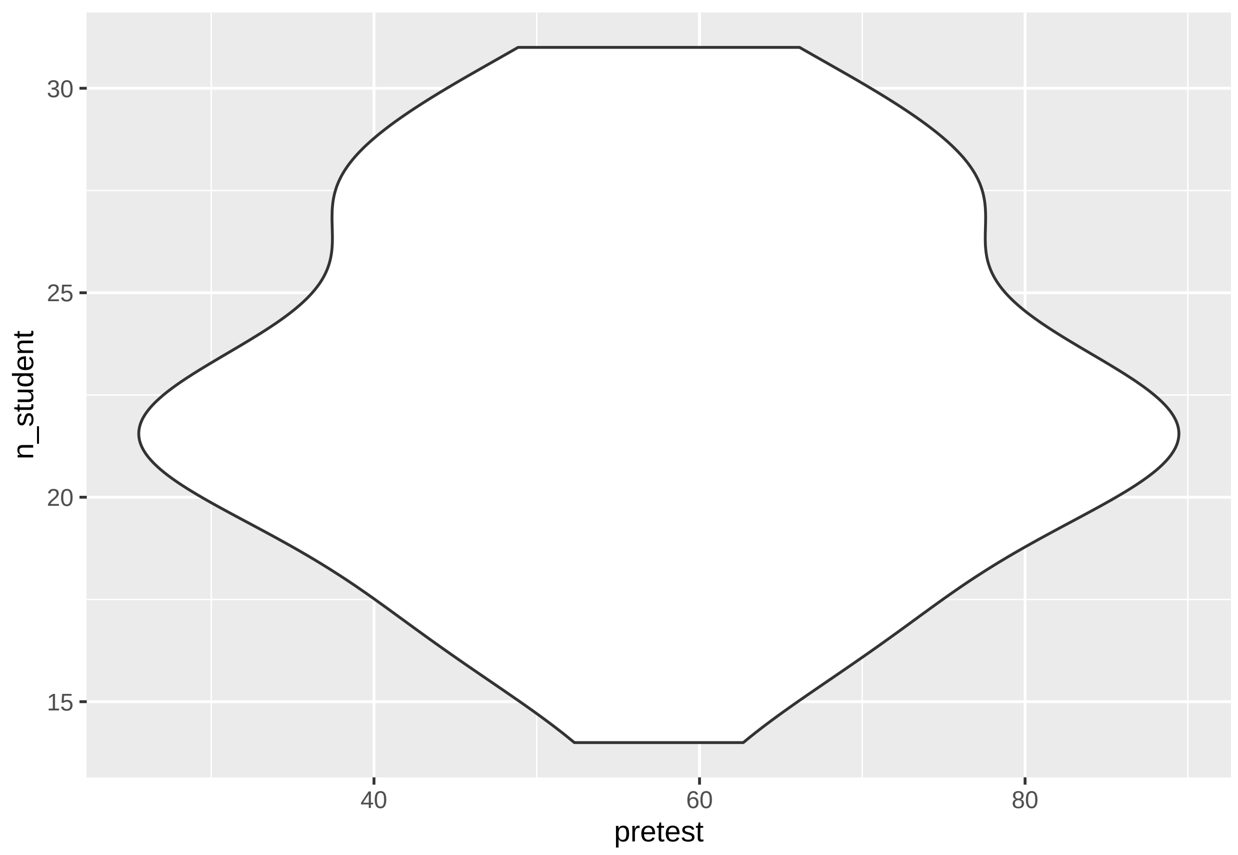
# Violin plot with area proportional to number of observations

```
data6 +geom_violin(scale = "count")
```

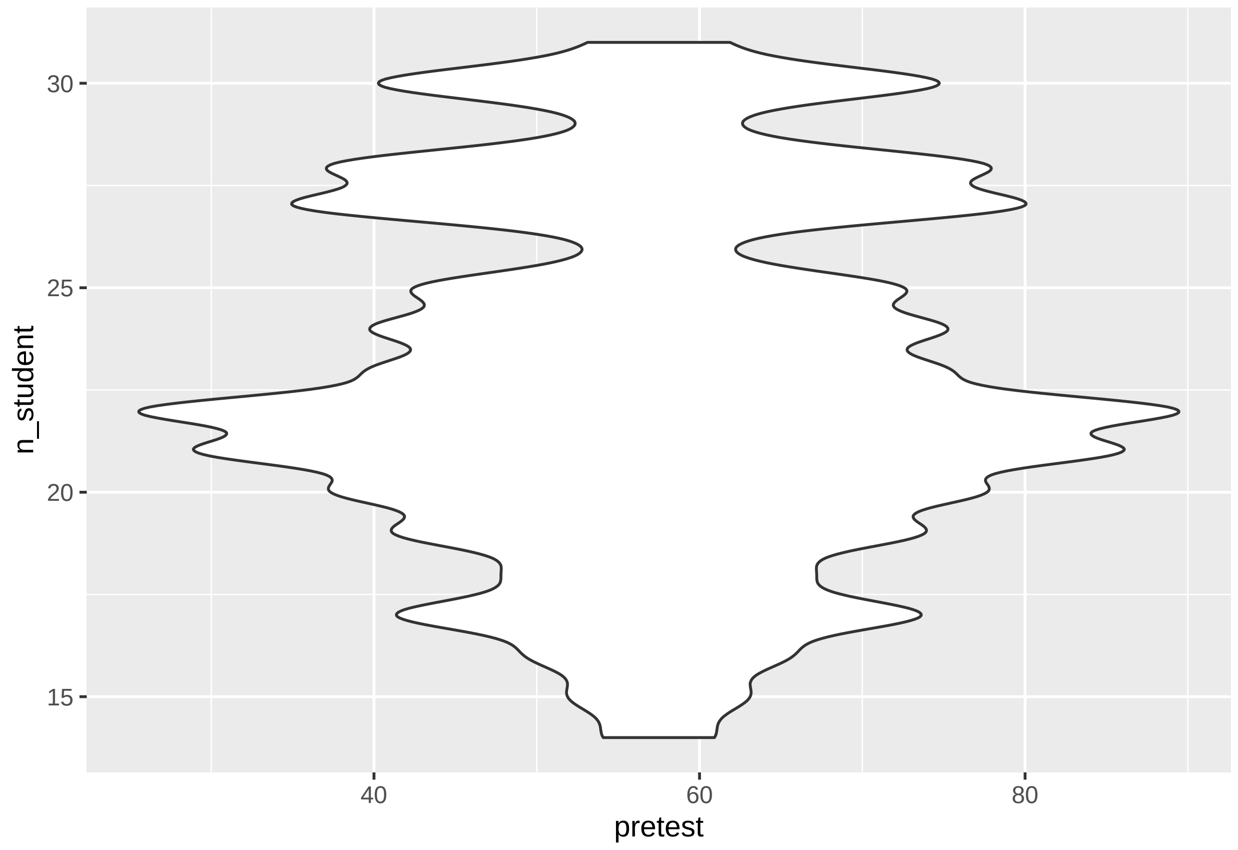


# Violin plot with

```
# More smoothing
data6+geom_violin(adjust = 2)
```



```
# Less smoothing
data6 +geom_violin(adjust = .5)
```

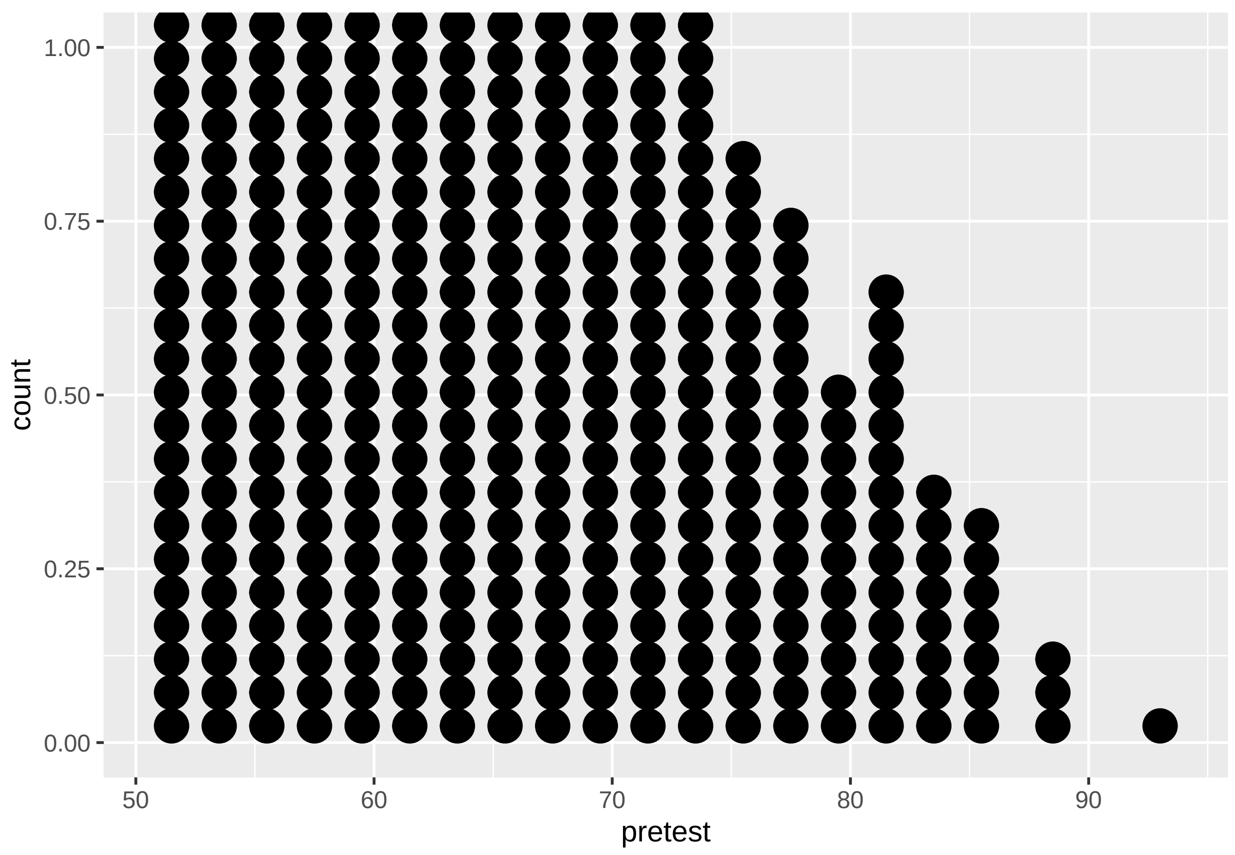


Making a Dot Plot

```
# Save a modified data set that only includes of males data for marks that contains > 50
data7 <- df %>%
filter(gender == "Male" & pretest > 50)
```

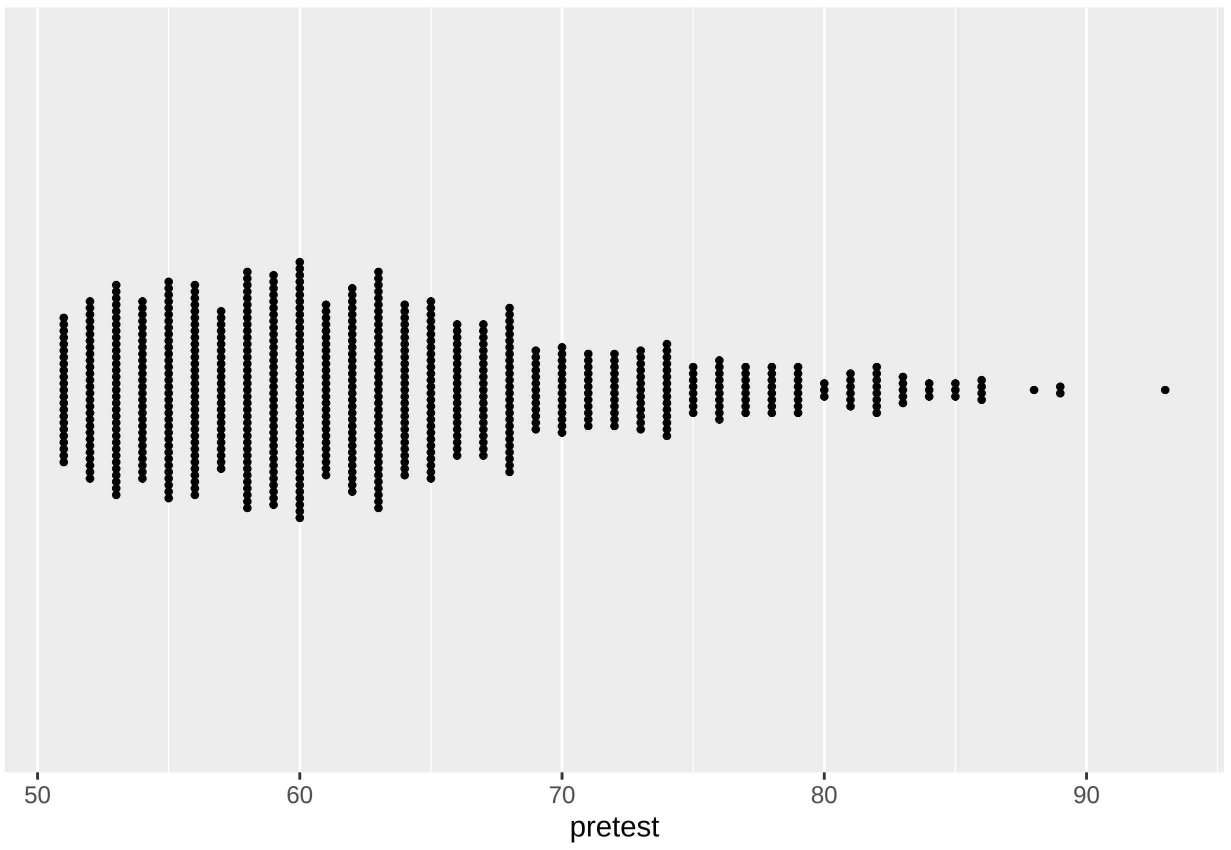
```
# Create a base ggplot object using `data7`, called `data7_p` (for data7 plot)
data8 <- ggplot(data7,aes(x=pretest))
data8+geom_dotplot()
```

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

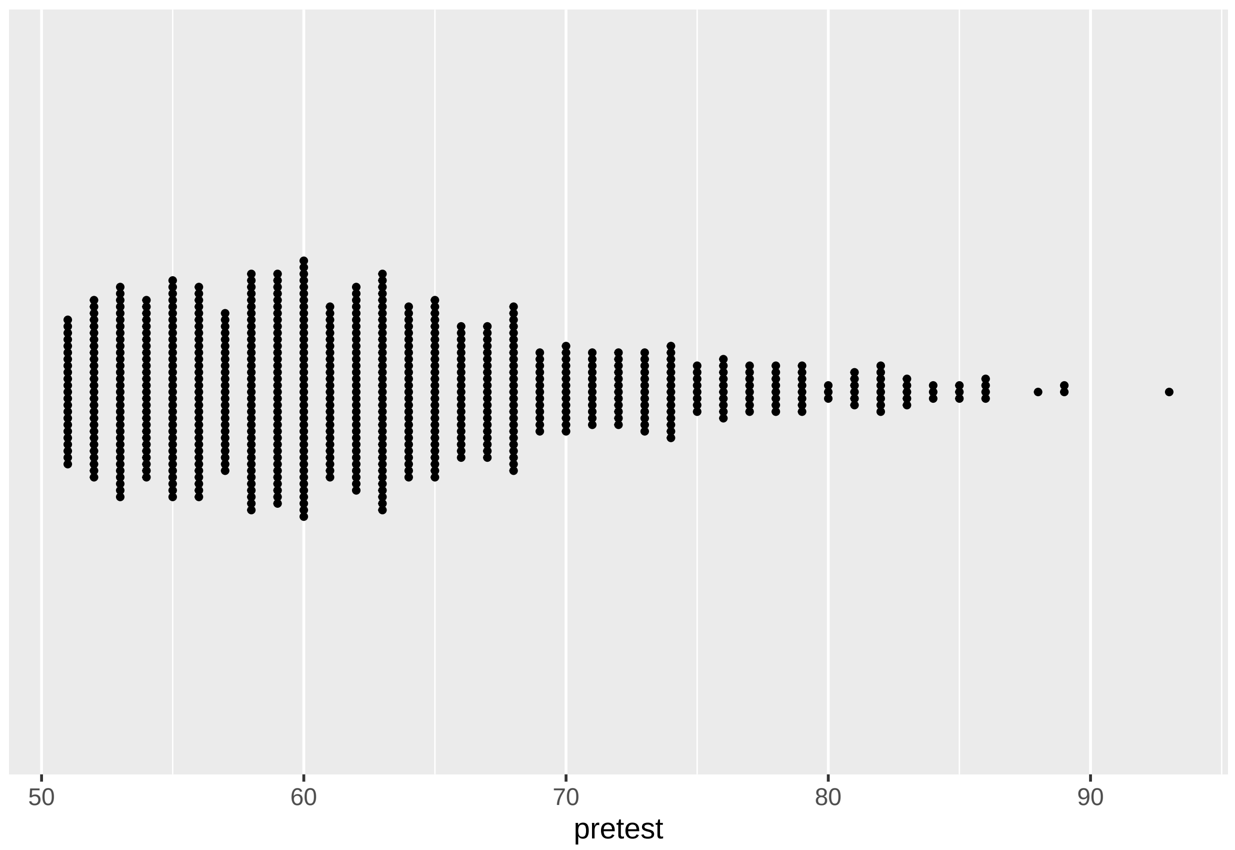


# Dot plot with stackdir = "center" (1) # With stackdir = "centerwhole"(2)

```
data8 +
geom_dotplot(binwidth = .25, stackdir = "center") +
scale_y_continuous(breaks = NULL) +
theme(axis.title.y = element_blank())
```

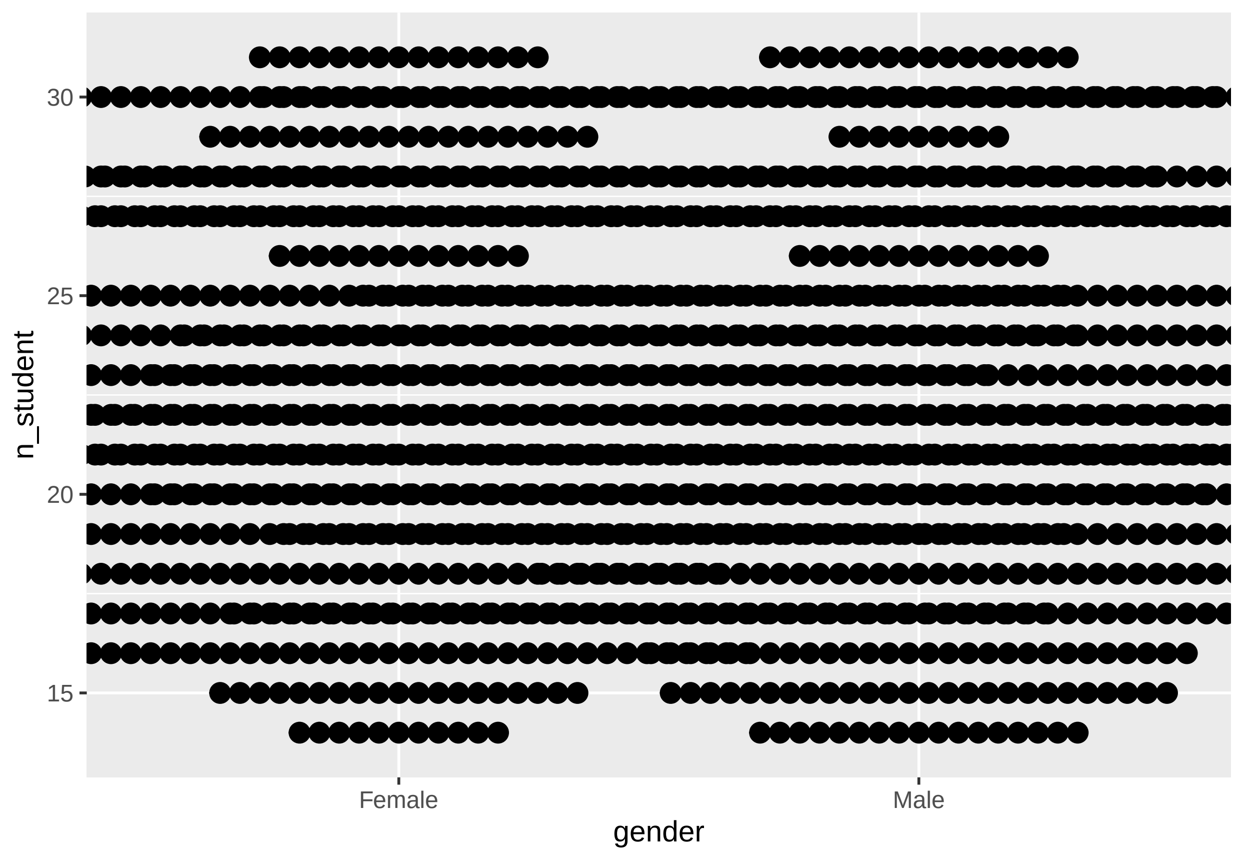


```
data8 +
geom_dotplot(binwidth = .25, stackdir = "centerwhole") +
scale_y_continuous(breaks = NULL) +
theme(axis.title.y = element_blank())
```



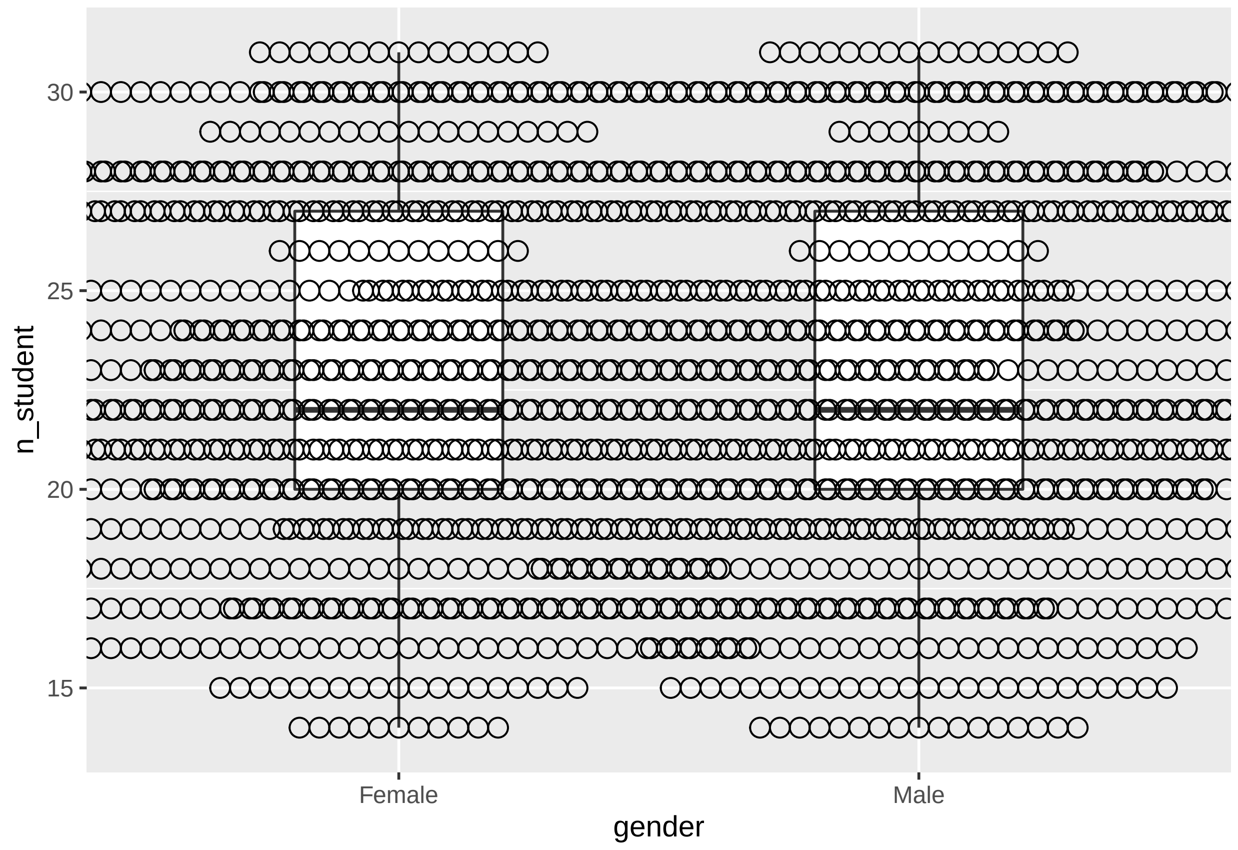
# Making Multiple Dot Plots for Grouped Data

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



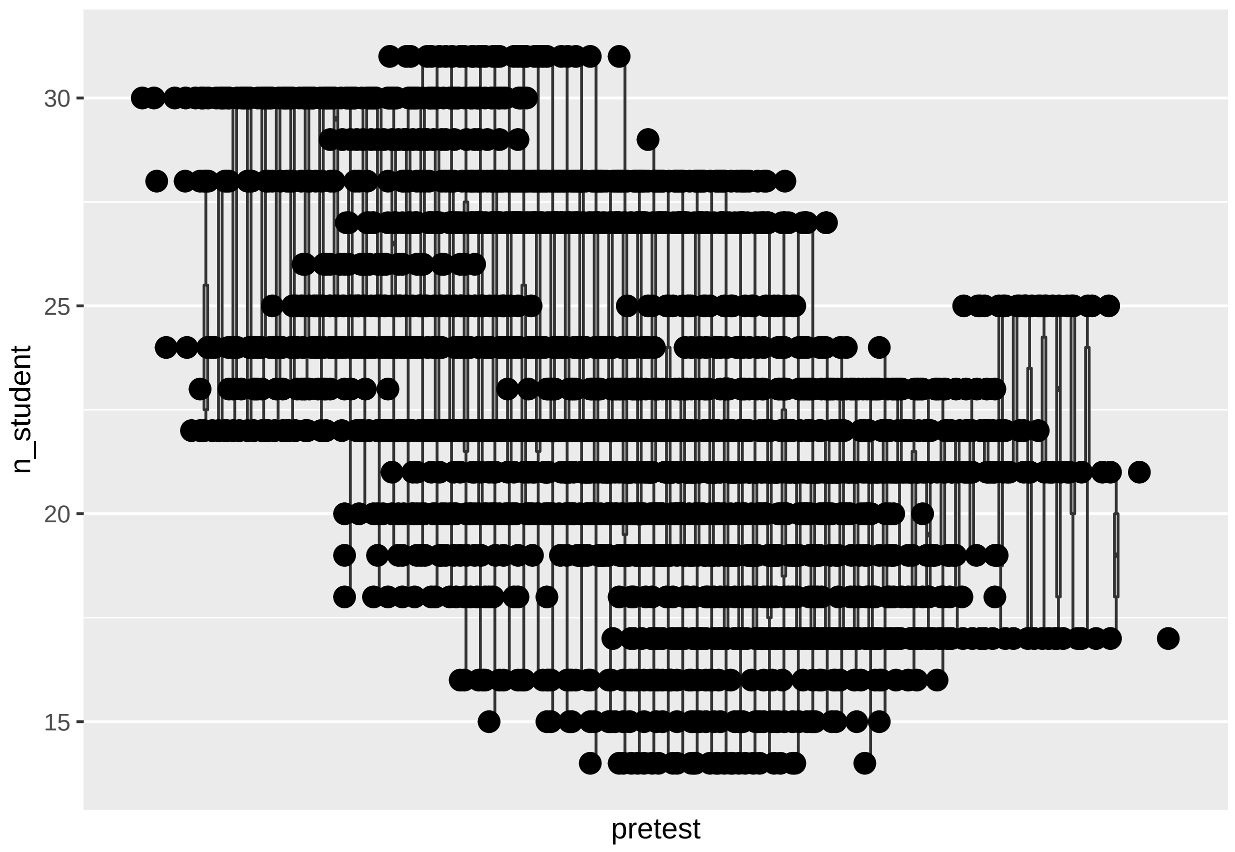
# Dot plot overlaid on box plot

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



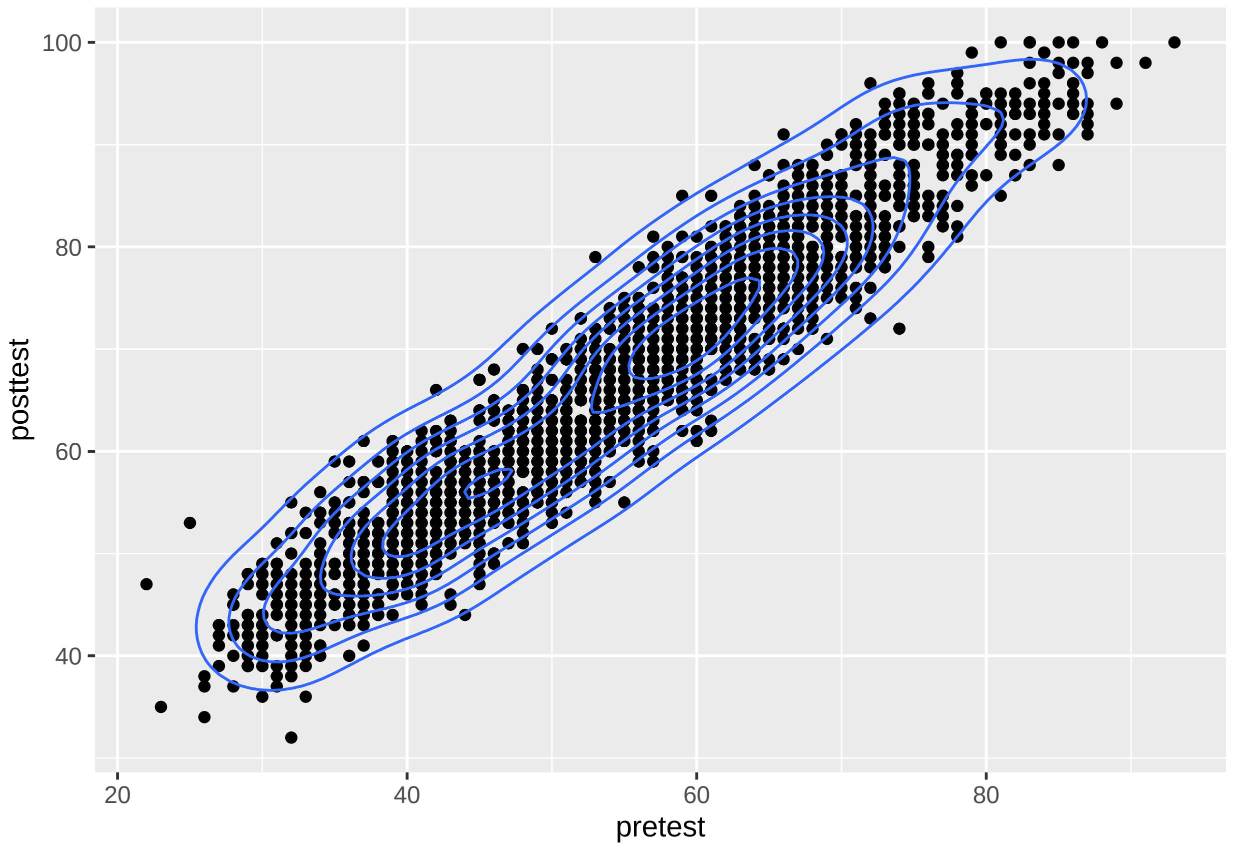
# Dot plot next to box plot

```
ggplot(df, aes(x = pretest, y = n_student)) +
geom_boxplot(aes(x = as.numeric(pretest) + .2, group = pretest), width = .25) +geom_dotplot(aes(x = as.n
)
```

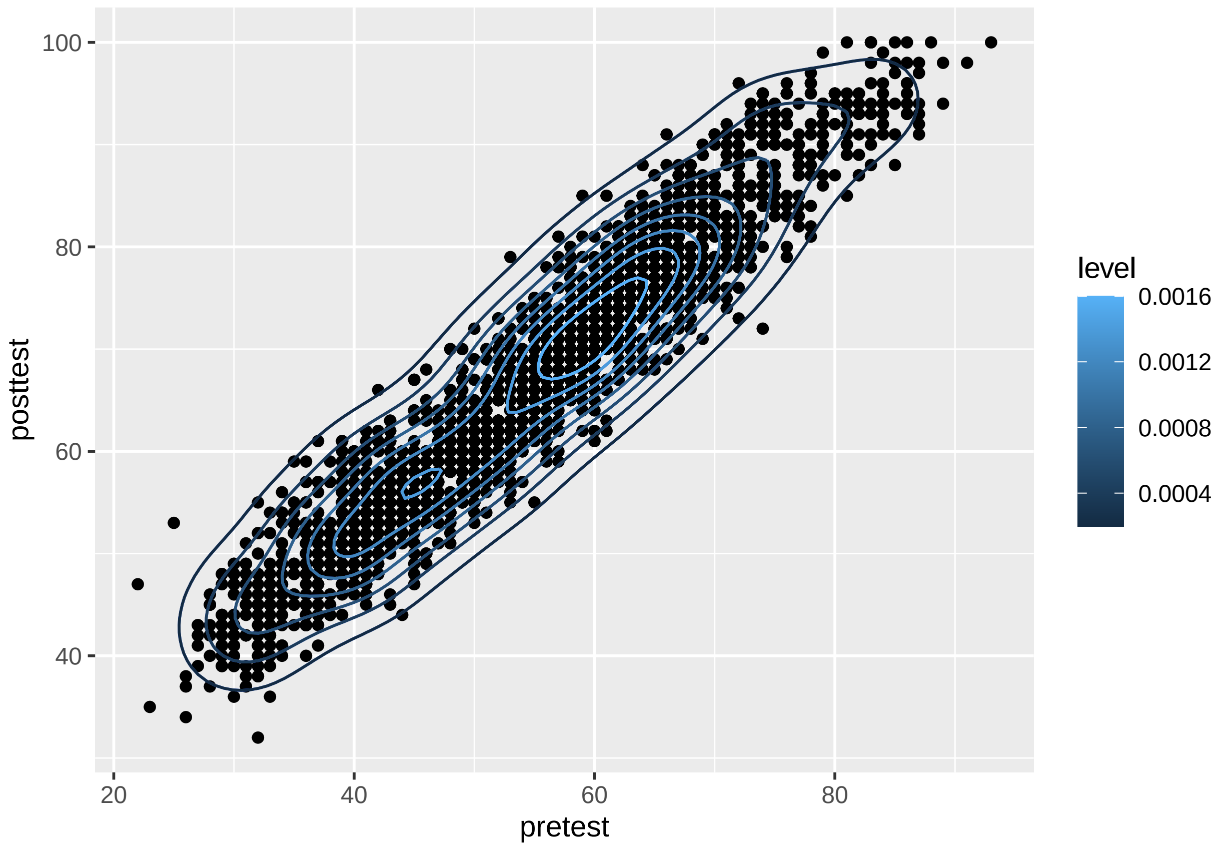


# Making a Density Plot of Two-Dimensional Data

```
# Save a base plot object
 ggplot(df, aes(x = pretest, y = posttest))+geom_point() +stat_density2d()
```

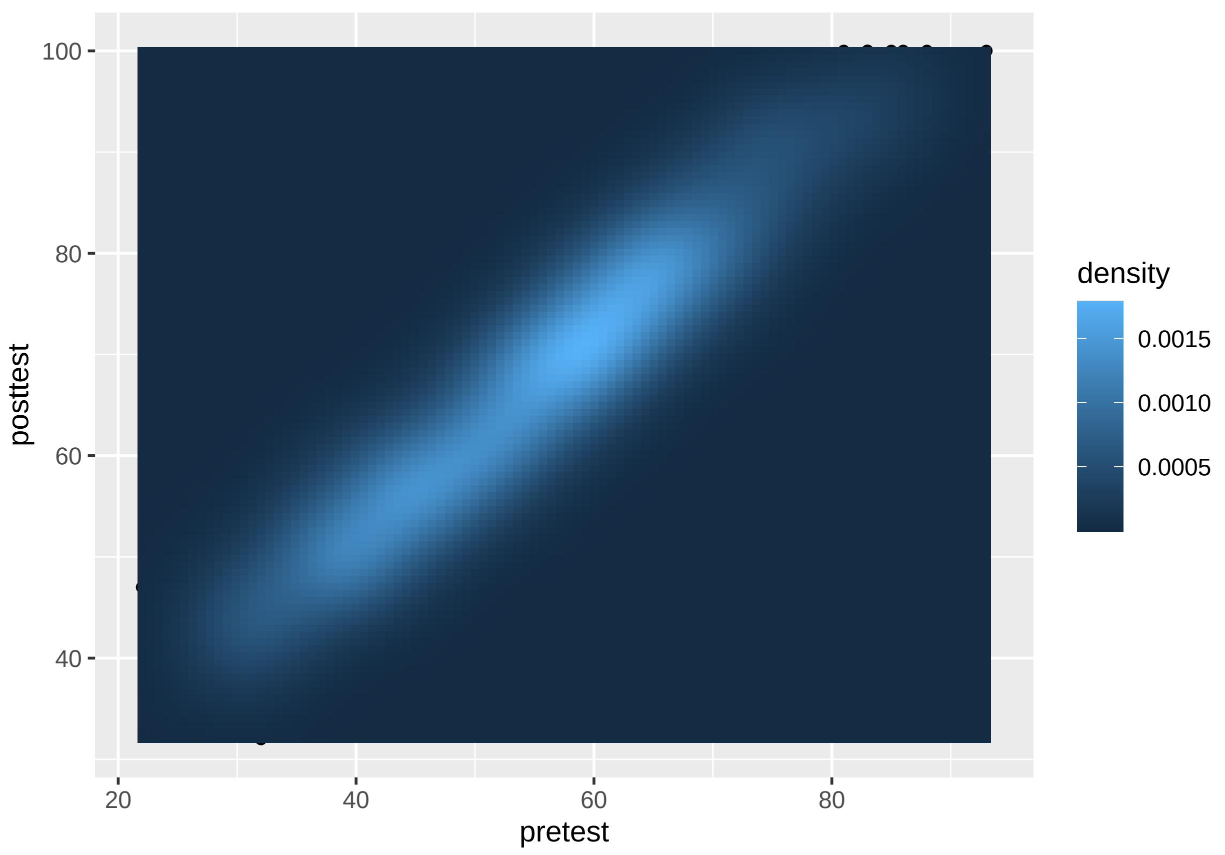


```
# Contour lines, with "height" mapped to color
ggplot(df, aes(x = pretest, y = posttest))+geom_point() +stat_density2d()+stat_density2d(aes(colour = .
```

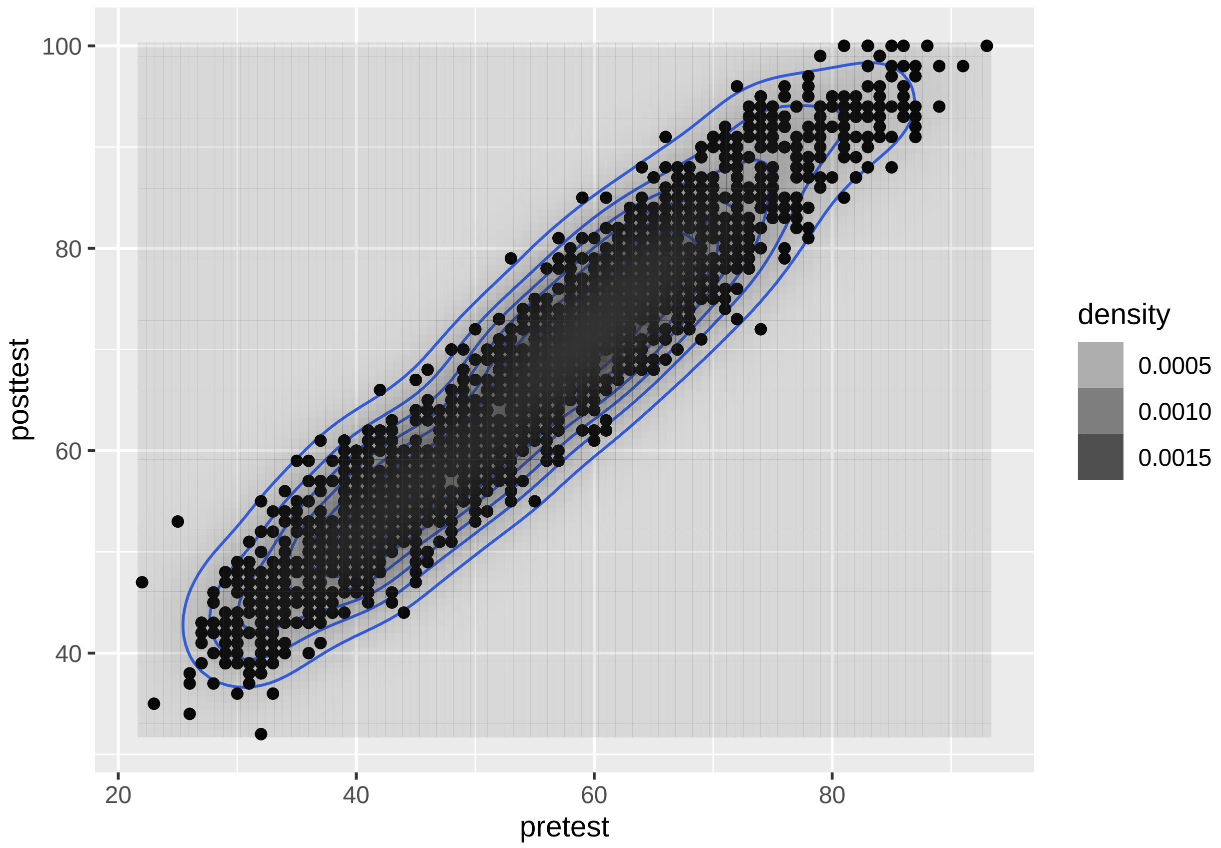


# With ..density.. mapped to fill (1) # With points, and ..density.. mapped to alpha(2)

```
# Map density estimate to fill color
ggplot(df, aes(x = pretest, y = posttest))+geom_point() +stat_density2d()+stat_density2d(aes(fill = ..de
```

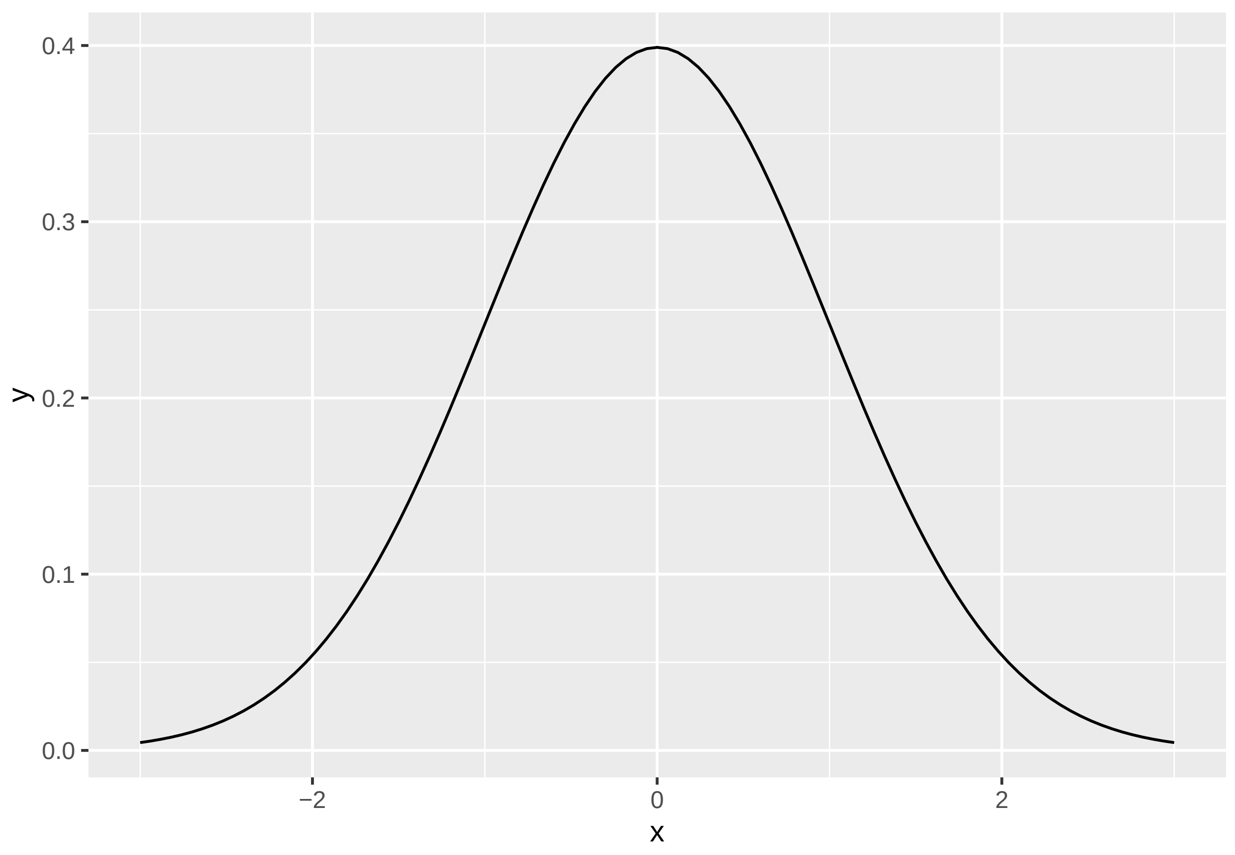


```
# With points, and map density estimate to alpha
ggplot(df, aes(x = pretest, y = posttest))+geom_point() +stat_density2d()+geom_point() +stat_density2d(a
```

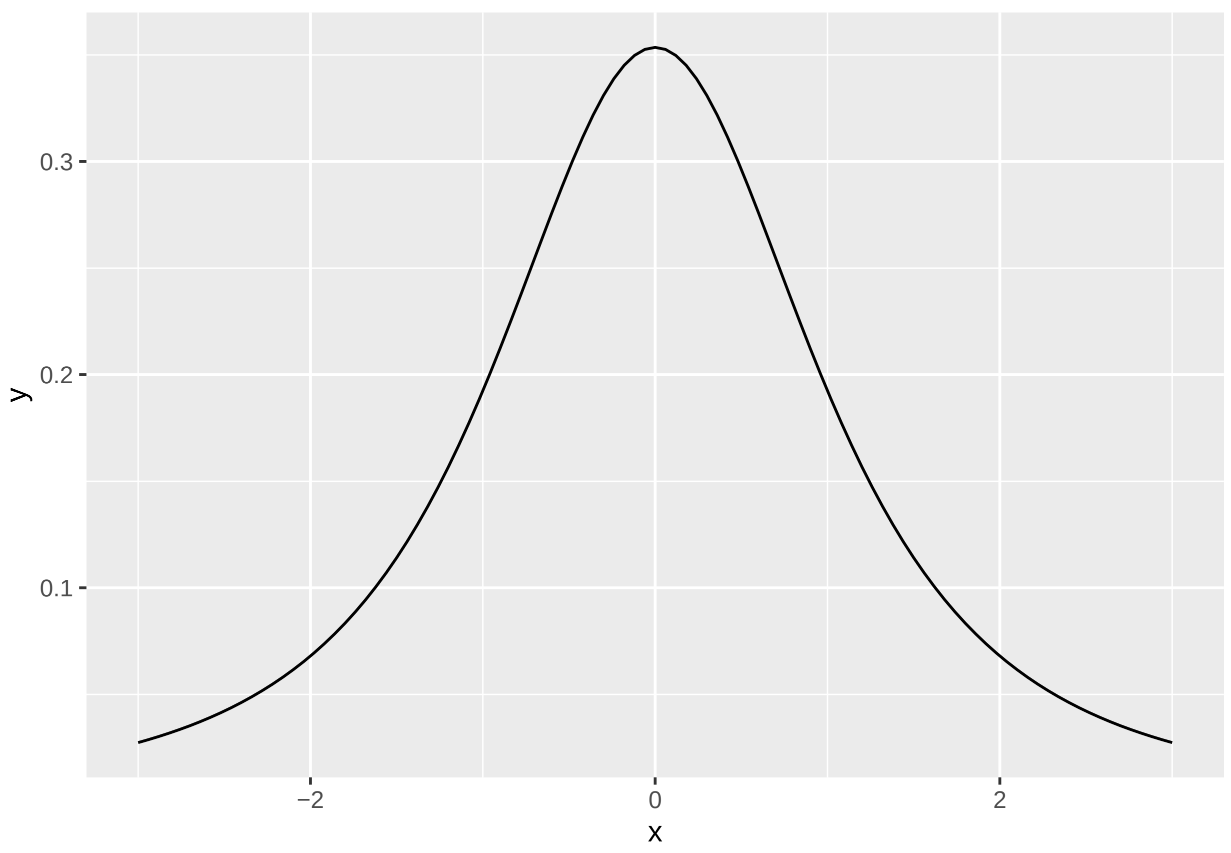


# Plotting a Function # # The data frame is only used for setting the range

```
# The normal distribution
ggplot(data.frame(x = c(-3, 3)), aes(x = x))  + stat_function(fun = dnorm)
```



```
# The t-distribution with df=2
ggplot(data.frame(x = c(-3, 3)), aes(x = x)) + stat_function(fun = dt, args = list(df = 2))
```


0.3
0.2
0.1
y
−2
0
2
x