

Stat 411/511

SOME BASIC R

Oct 9 2015

Today

- * **Some R** (to make sure you all see it before lab)
 - * **Subsetting**
 - * **Basic plotting with ggplot2**

you've seen this a few times

```
library(Sleuth3)
```

\$ubsetting

The \$ lets you get columns out of a data.frame

```
case0101$Score
```

```
case0101$Treatment
```

```
mean(case0101$Score)
```

```
sd(case0101$Score)
```

```
table(case0101$Treatment)
```


Subsetting rows

The function `subset` is useful for getting specific rows in a `data.frame`.

E.g.

```
subset(case0101, Treatment == "Extrinsic")
```


data.frame


logical statement
evaluates to TRUE or FALSE
for each row

Returns another `data.frame`

Logical statements

== equal

!= not equal

> greater than

< less than

>=

<=

! not

& and

| or

%in%

```
x <- 1:5  
y <- c("a", "a", "b", "b", "c")
```

```
x == 2
```

```
x > 2
```

```
x != 2
```

```
y == "a"
```

```
y == "a" & x == 2
```

```
subset(y, x == 2)
```

```
y %in% c("b", "c")
```

TRUE, FALSE or NA

```
> pets
  pet_name  type  owner_name  age
1  Dexter   cat   Charlotte   3
2  Scylla   cat   Charlotte   2
3  Medusa   cat           Josh   11
4   Mina     dog   Hadley     5
```

Write subset commands that will return:

1. Charlotte's pets
2. pet's that don't belong to Charlotte
3. cats
4. cats that are older than 2
5. dogs that are younger than 4

Combine them with \$ and mean to find:

1. The average age of Charlotte's pets
2. The average age of pet's that don't belong to Charlotte

[, general subsetting

give it:

- * logical
- * positive numbers
- * negative numbers
- * empty

See R cookbook (link given in lab)

To master R you should aim to master subsetting with [
For 411/511 master subset

Plotting with ggplot2

```
library(ggplot2)
```

Variable on the y-axis
(vertical axis)

```
ggplot(x = Treatment, y = Score,  
data = case0101)
```

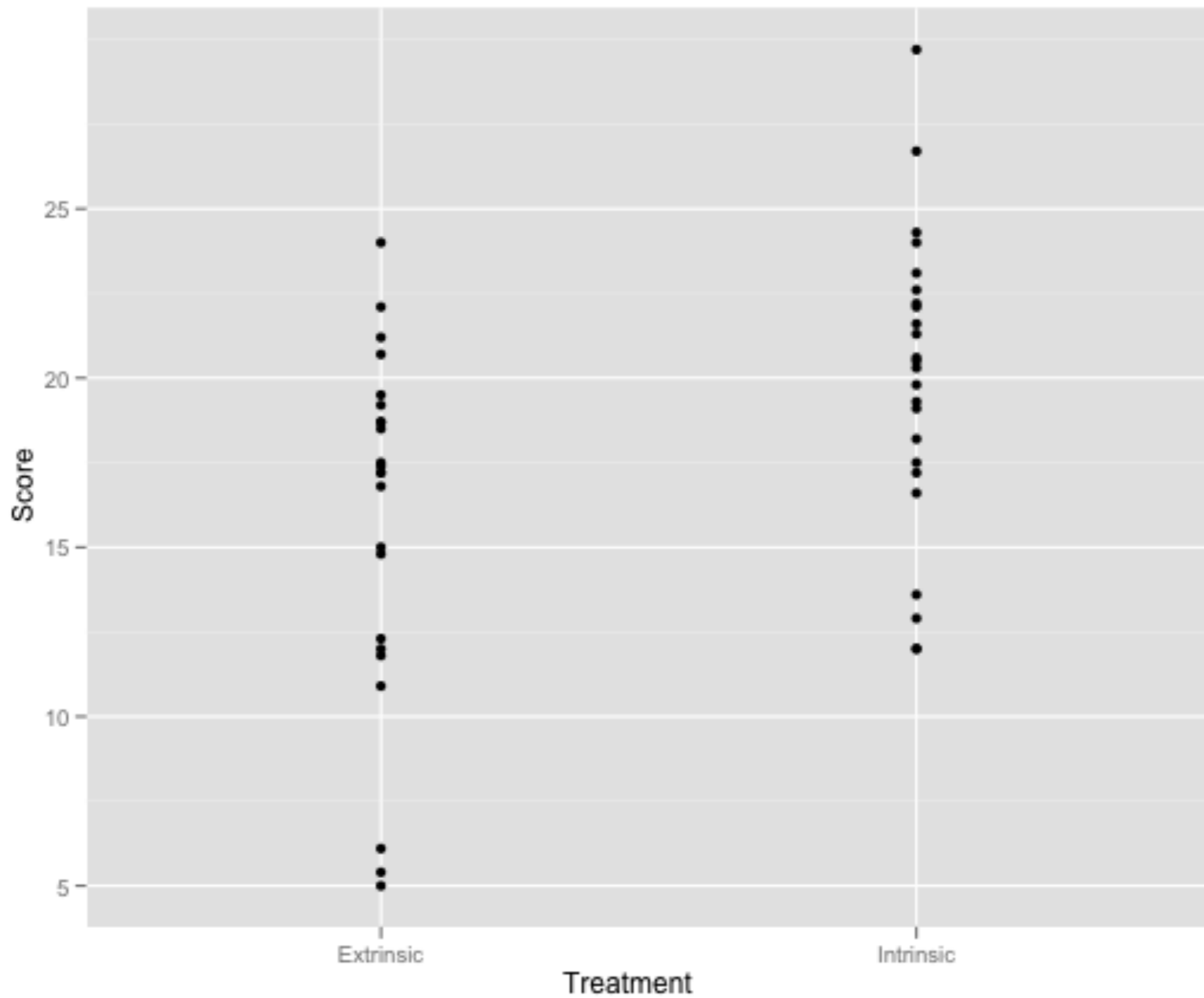
Variable on the x-axis
(horizontal axis)

Data.frame the
variables are in

```
ggplot(Treatment, Score, data = case0101)
```

This is the short hand version

```
qplot(Treatment, Score, data = case0101)
```



Geometric objects

Points are just one type of geometric object (and the default in `qplot`)

```
qplot(Treatment, Score, data = case0101,  
      geom = "point")
```

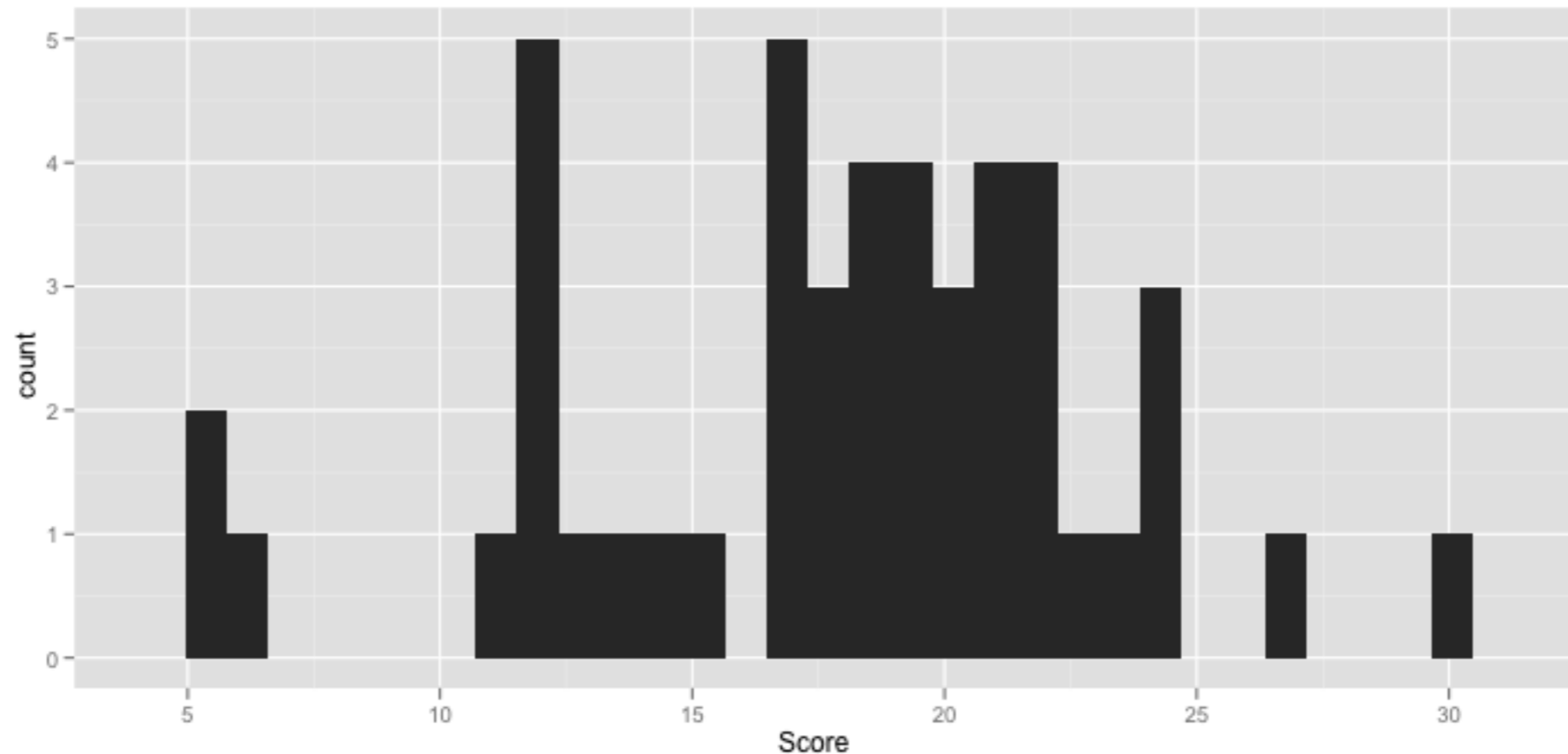
```
qplot(Treatment, Score, data = case0101,  
      geom = "jitter")
```

points with random
jitter in both x and y

```
qplot(Treatment, Score, data = case0101,  
      geom = "boxplot")
```

boxplots for each unique x value
(if x is categorical)

Histograms



Which variable is on the y-axis of a histogram?

```
qplot(Score, data = case0101,  
      geom = "histogram")
```

Facetting

Facetting produces the same plot for different subsets of the data

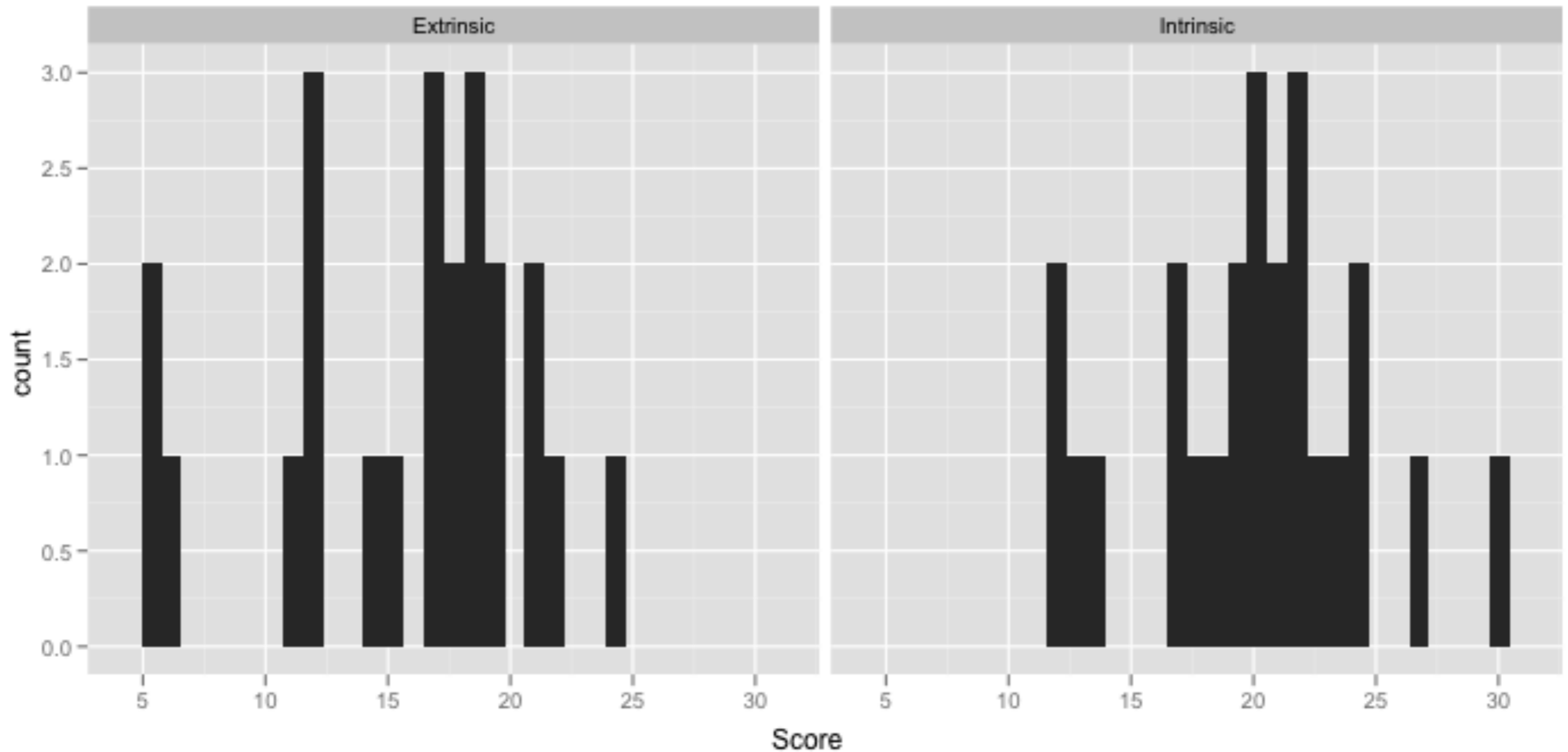
```
qplot(Score, data = case0101,  
      geom = "histogram") + ← typed
```

```
facet_wrap( ~ Treatment)
```

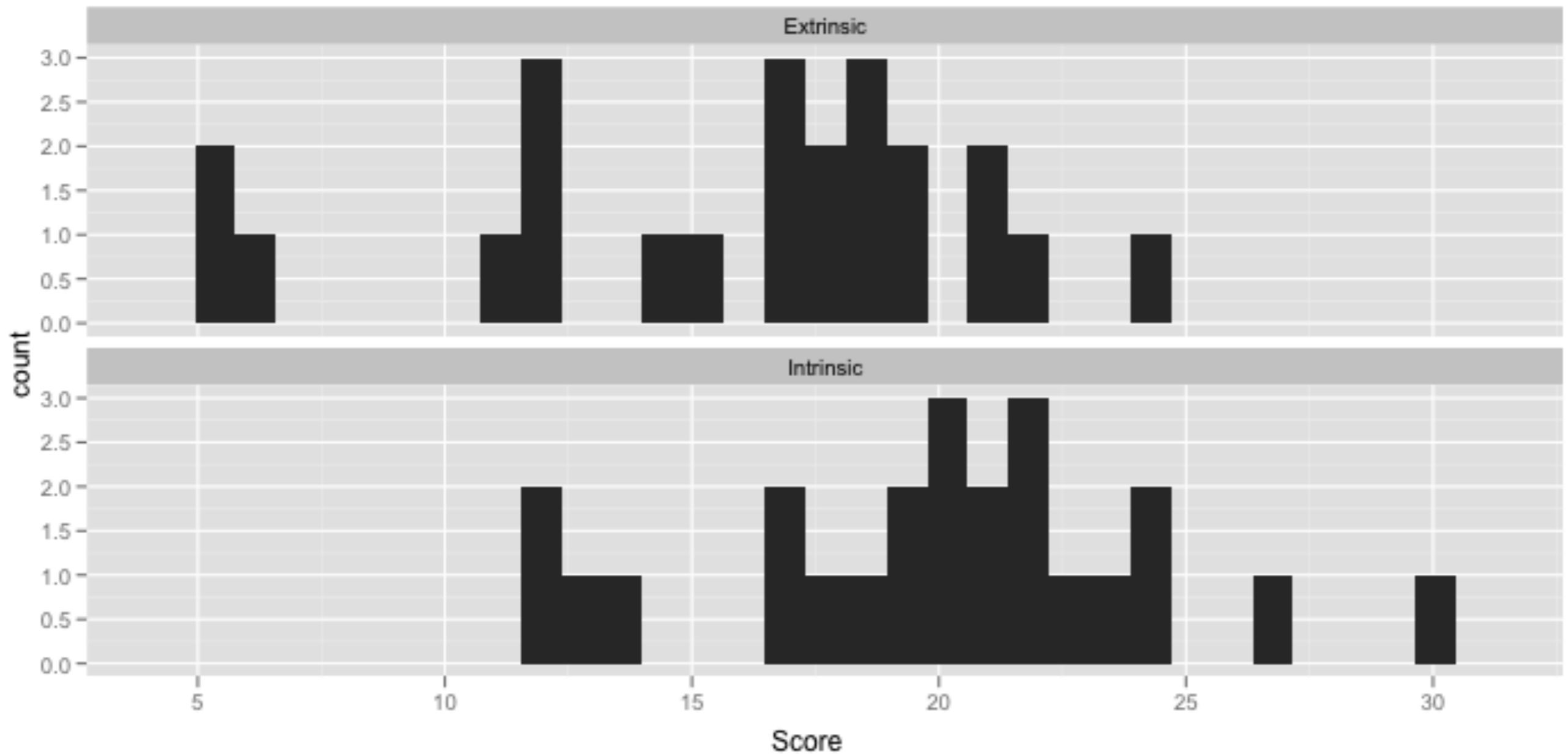
A facet for each
unique value of Treatment

the name of
another column in data

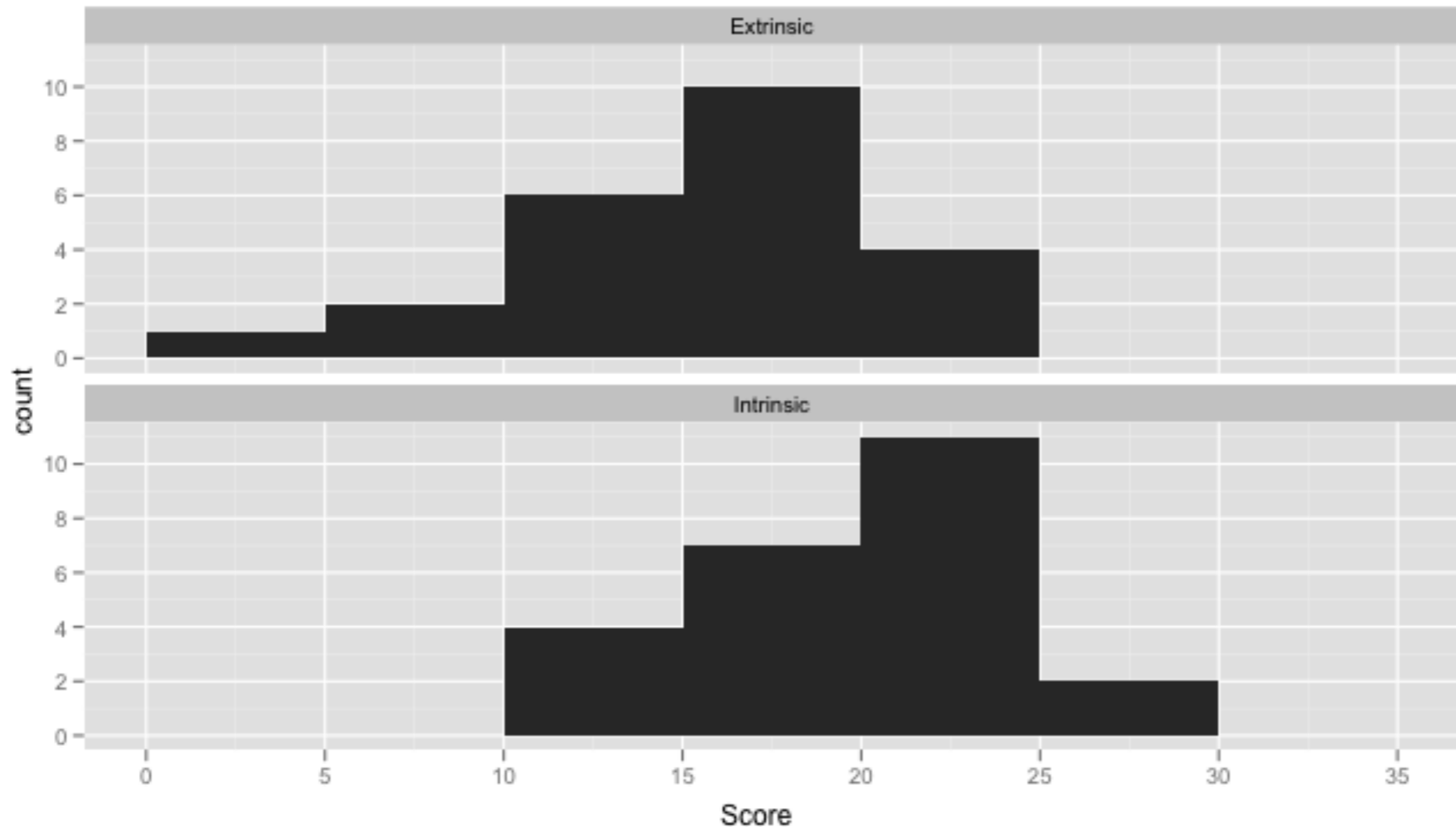
```
qplot(Score, data = case0101,  
      geom = "histogram") +  
      facet_wrap( ~ Treatment)
```



```
qplot(Score, data = case0101,  
      geom = "histogram") +  
      facet_wrap( ~ Treatment, ncol = 1)
```



```
qplot(Score, data = case0101,  
      geom = "histogram", binwidth = 5) +  
      facet_wrap( ~ Treatment, ncol = 1)
```




```
> head(case0202)
  Unaffected Affected
1      1.94      1.27
2      1.44      1.63
3      1.56      1.47
4      1.58      1.39
5      2.06      1.93
6      1.66      1.26
```

**How could you
reproduce this plot?**

