wk2: Describing Data Graphically and Numerically

D. ODay

2022-06-21

Displaying Data Through Time

Problem

An engineer gathered 20 consectutive computer ans from a production line, keeping track of the order in which the fans were produced. Then these fans were tested for airflow and cubic feet per minute. The testing produced the following:

Fans 1-10: `68, 72, 72, 74, 72, 69, 75, 75, 72, 73`
Fans 10-20: `70, 71, 71, 72, 73, 72, 70, 72, 73, 74`

A run chart measures something over time

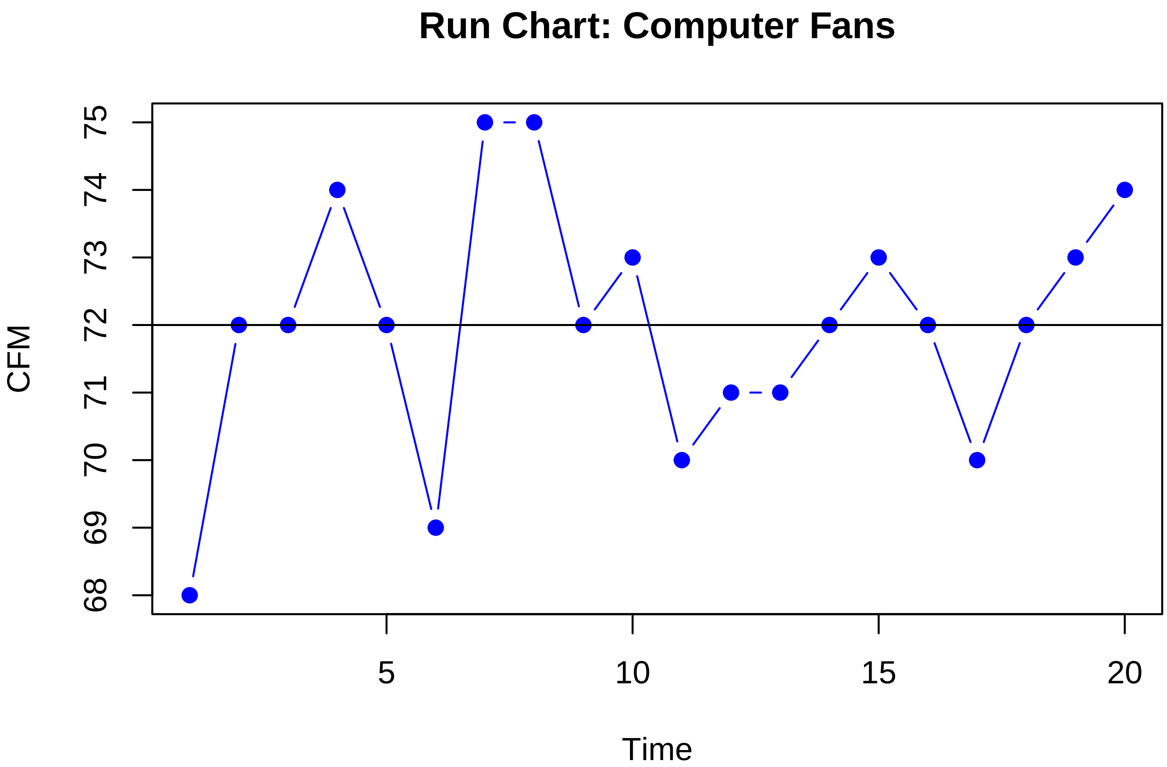
How to do it in R:

```
# Create a vector
cfm = c(68, 72, 72, 74, 72, 69, 75, 75, 72, 73, 70, 71, 71, 72, 73, 72, 70, 72, 73, 74)

#Store the data in a dataframe
fans = data.frame(cfm)


#Create the run chart
require(lolcat)
```

```
## Loading required package: lolcat
```

```
## lolcat 2.0.0
```

```
spc.run.chart(fans$cfm, main="Run Chart: Computer Fans", ylab="CFM")

#Add a Horizontal Line of the Average
mean_cfm = mean(fans$cfm)
abline(h=mean_cfm)
```



Other Options to Use for Customization

- Point Symbol : pch (1-25)
- Point Size : cex=
- Color = "color_name"
- Line type : lty = (0-6)
- Line width: lwd =

Frequency Visualizations

Frequency Distributions

Frequency distributions provide us with a method for arranging and viewing data sets. This allows for easier interpretation and analysis of data.

Ungrouped vs Groupd Frequency distributions Use ungrouped when there are less than 20 unique data values in the dataset.

Use grouped when there are more than 20 unique data values in the data set.

we'll use the same frequency values as before

Our distribution will list - Value - Frequency - Relative Frequency - Cumulative up/down

Frequency distributions are considered 'ungrouped' when each row, or 'class interval', consists of onle one score, value or observation

When the range of the dataset is large, constructing a functional ungrouped frequency distribution becomes untenable. We'd use a grouped frequency distribution.

Grouped frequency distributions have a range of values associated with each interval

```r
#Ungrouped Frequency Distribution
frequency.dist.ungrouped(fans$cfm)
```

```
##   value freq rel.freq cum.up cum.down
## 1    68    1     0.05   0.05     1.00
## 2    69    1     0.05   0.10     0.95
## 3    70    2     0.10   0.20     0.90
## 4    71    2     0.10   0.30     0.80
## 5    72    7     0.35   0.65     0.70
## 6    73    3     0.15   0.80     0.35
## 7    74    2     0.10   0.90     0.20
## 8    75    2     0.10   1.00     0.10
```

```r
#Grouped Frequency Distribution
castings <- read.csv("~/Documents/GitHub/school_cu/school_cu/methods for quality improvement/DTSA5704_D


frequency.dist.grouped(castings$weight)
```

```
##    l min midpoint max u freq rel.freq cum.up cum.down
## 1  [ 105    107.5 110 )    1    0.025  0.025    1.000
## 2  [ 110    112.5 115 )    1    0.025  0.050    0.975
## 3  [ 115    117.5 120 )    2    0.050  0.100    0.950
## 4  [ 120    122.5 125 )    6    0.150  0.250    0.900
## 5  [ 125    127.5 130 )    8    0.200  0.450    0.750
## 6  [ 130    132.5 135 )    6    0.150  0.600    0.550
## 7  [ 135    137.5 140 )    4    0.100  0.700    0.400
## 8  [ 140    142.5 145 )    2    0.050  0.750    0.300
## 9  [ 145    147.5 150 )    3    0.075  0.825    0.250
## 10 [ 150    152.5 155 )    1    0.025  0.850    0.175
## 11 [ 155    157.5 160 )    3    0.075  0.925    0.150
## 12 [ 160    162.5 165 )    1    0.025  0.950    0.075
## 13 [ 165    167.5 170 )    1    0.025  0.975    0.050
## 14 [ 170    172.5 175 )    1    0.025  1.000    0.025
```

Rule of Thumb: generate a frequency distribution with as close as you can get to 10 class intervals without going under 10. So, just divide the range by 10.

Start the first class interval with a number that is a multiple of the class interval size

The first class interval must contain the lowest score in the data set.

Frequency Polygons and Histograms

- Useful For
  - Evaluating a manufacturing or business process
  - Determining machine and process capabilities
  - Comparing material, vendor, operator,process and product characteristics

Frequency Polygon

A graph or chart which represents the frequency of observations at each class interval(grouped) or value/score (ungrouped).

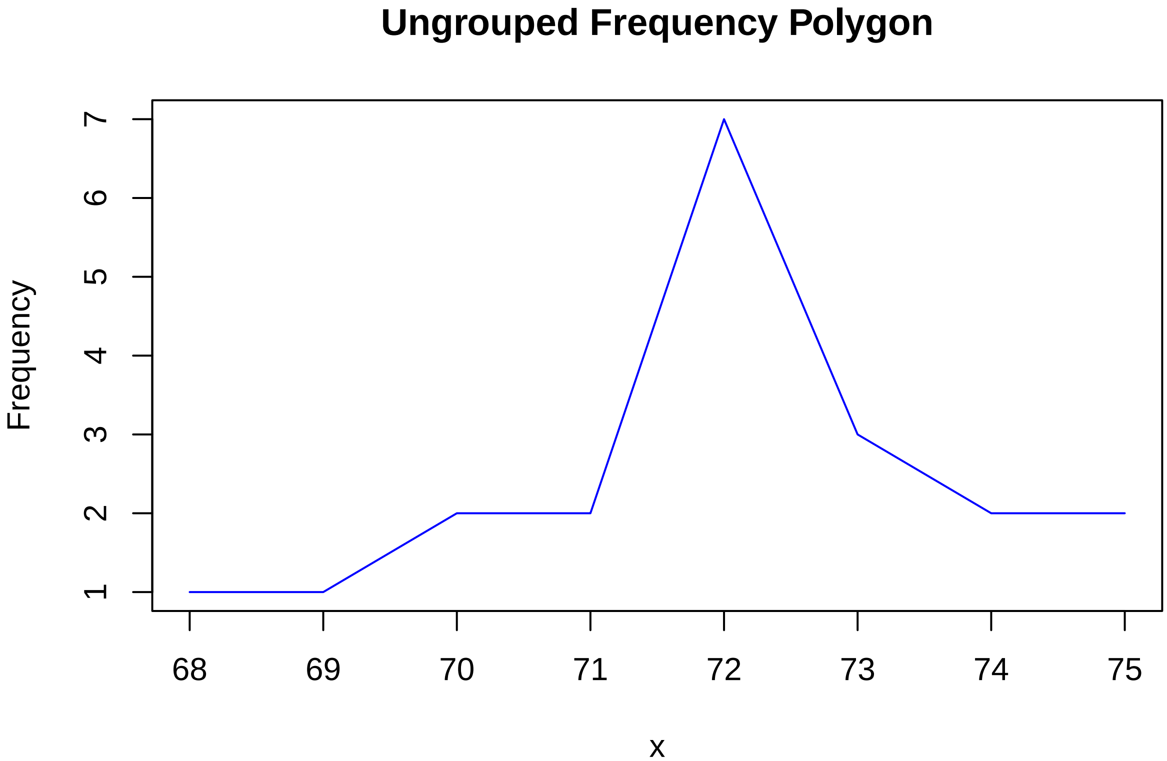
Similar to the frequency column of the frequency distribution

Frequency polygons often present

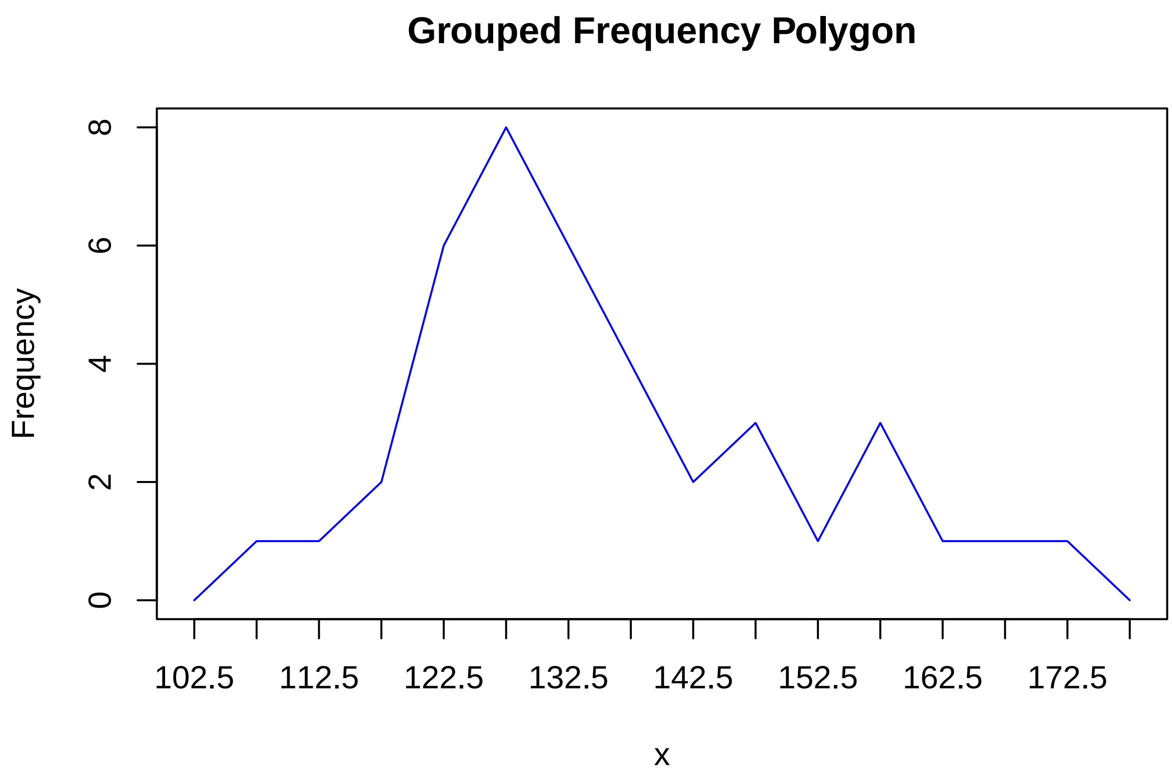
Advantages Often present a more representative illustration of the data pattern when data are measured along a continuous scale.

The polygon becomes increasingly smooth and curve-like as the number of class intervals and sample size increases, more closely representing the sampled population.

```
#Ungrouped Frequency Polygon
frequency.polygon.ungrouped(fans$cfm)
```




```
#Grouped Frequency Polygon
frequency.polygon.grouped(castings$weight)
```

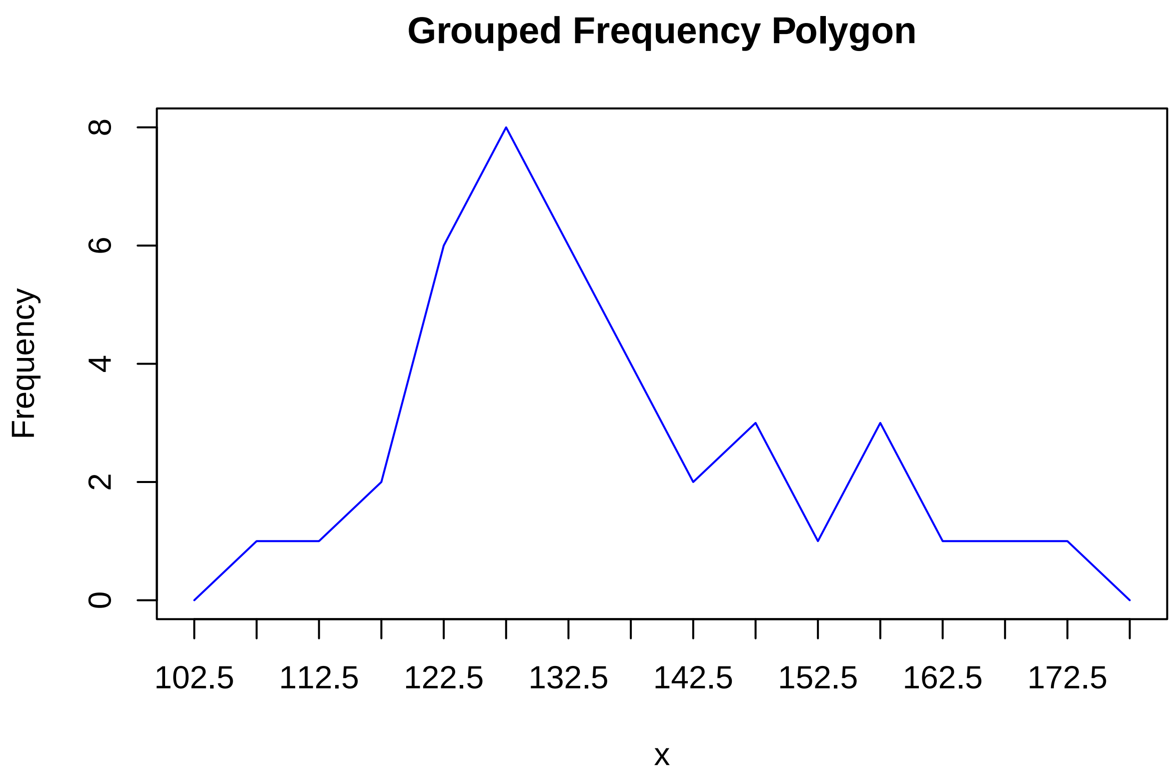


Histogram

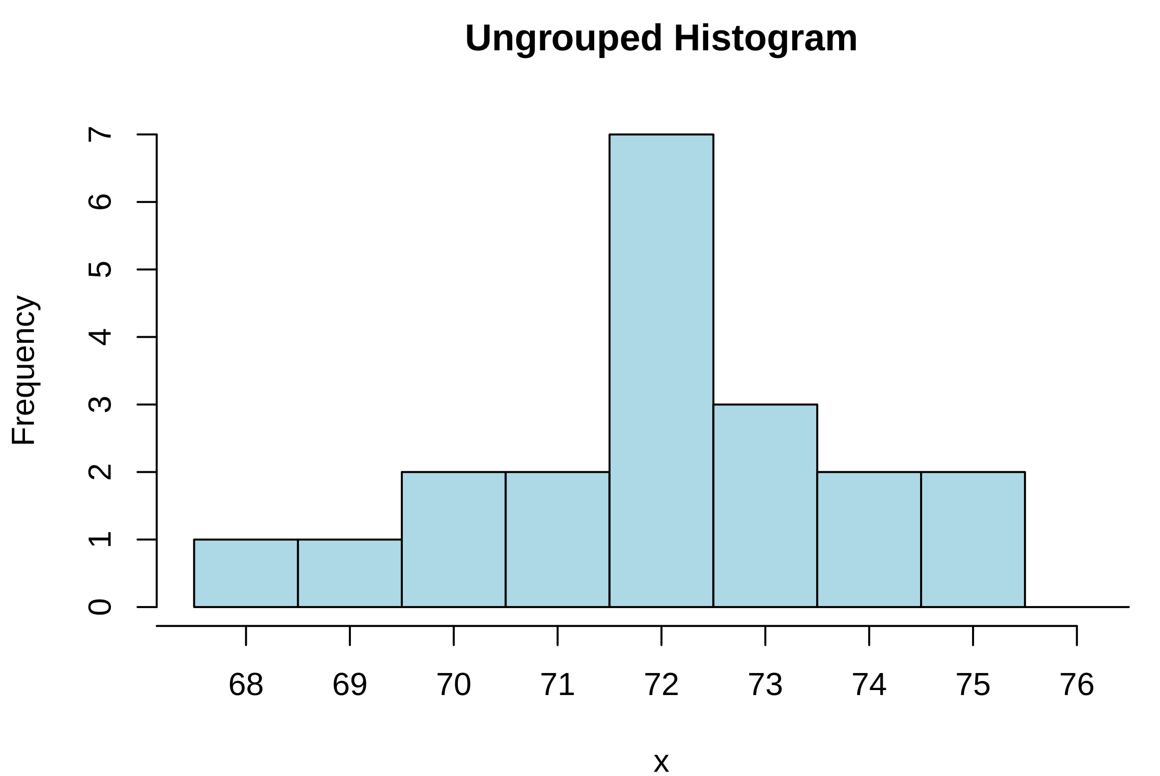
Similar to frequency polygon except bars are used

- When to use which:
  - a histogram when the values are discrete
  - a polygon or histogram when continuous

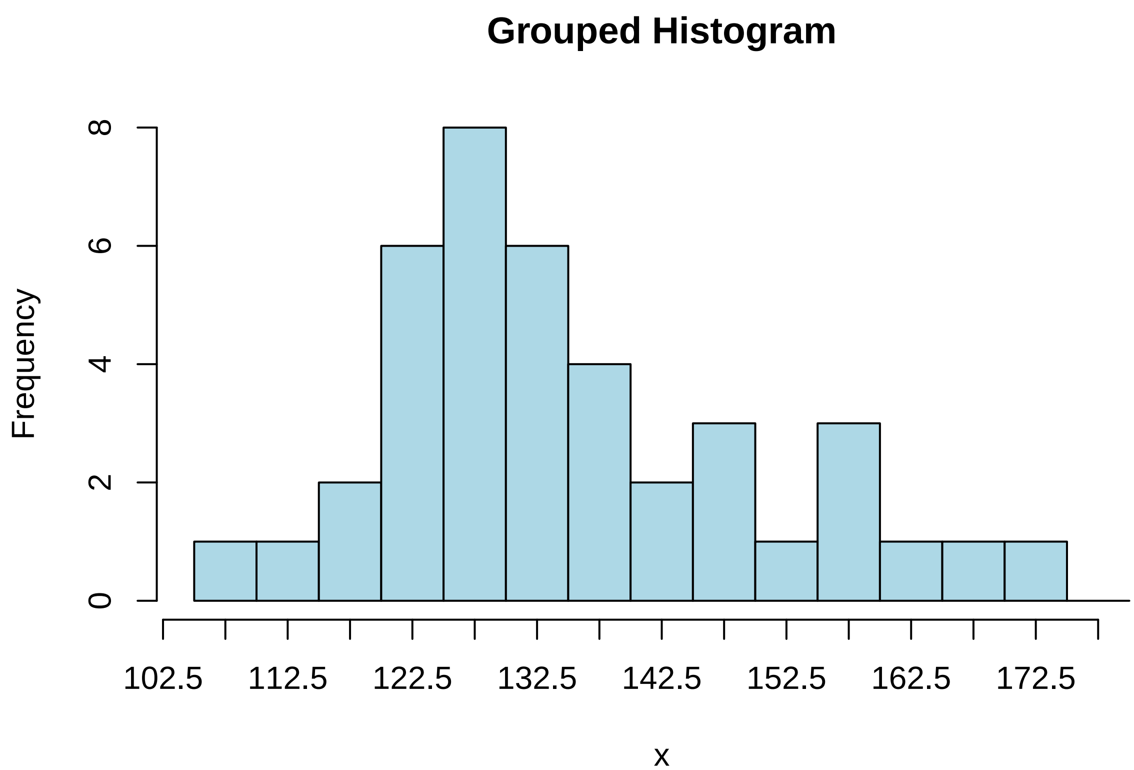
```
#Grouped Frequency Polygon
frequency.polygon.grouped(castings$weight)
```




```r
# Ungrouped Histogram
hist.ungrouped(fans$cfm)
```



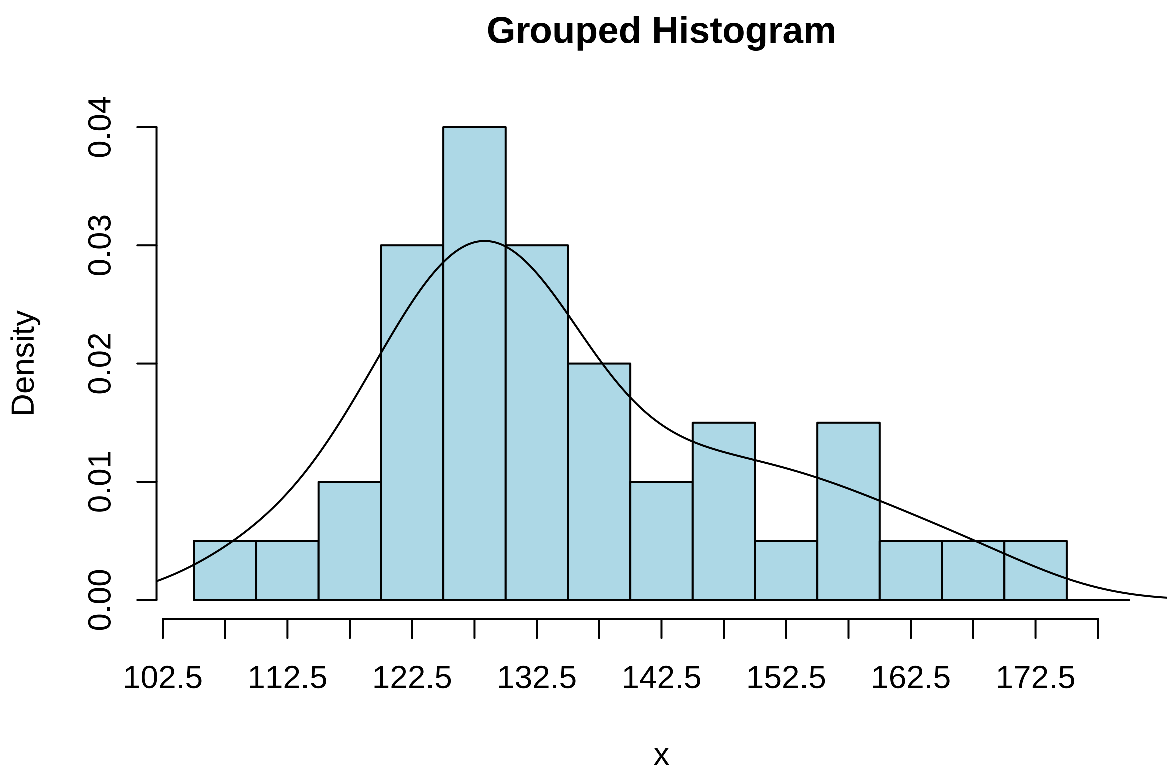
```r
# Grouped Histogram
hist.grouped(castings$weight)
```



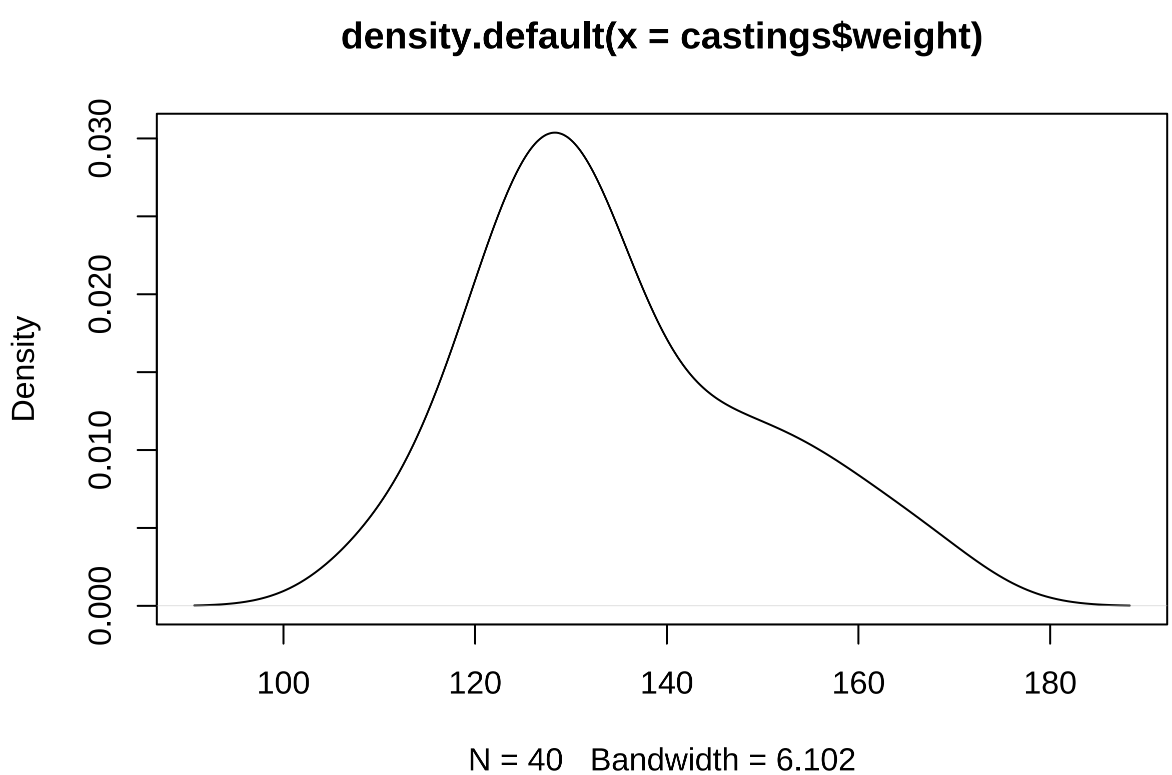
Histogram Patterns and Density Plots

Density plots are used with continuous data and can be used over a histogram.

```r
#Histogram with Density Line
hist.grouped(castings$weight, freq=F)
lines(density(castings$weight))
```




```
#Just density
plot(density(castings$weight))
```



Box and Whisker Plots

Display data corresponding to Percentiles, and typically from two or more sources or process streams simultaneously.

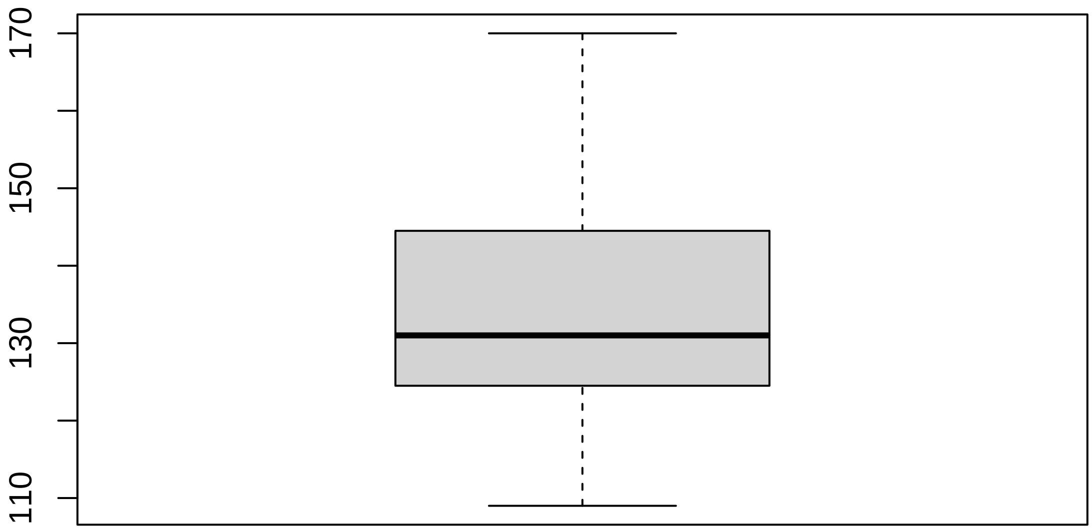
An advantage is that the two sample data sets do not have to posses the same shape but are directly comparable. Also, can display outliers.

A **notched box and whiskers plot** shows a 95% confidence intercal of the median.

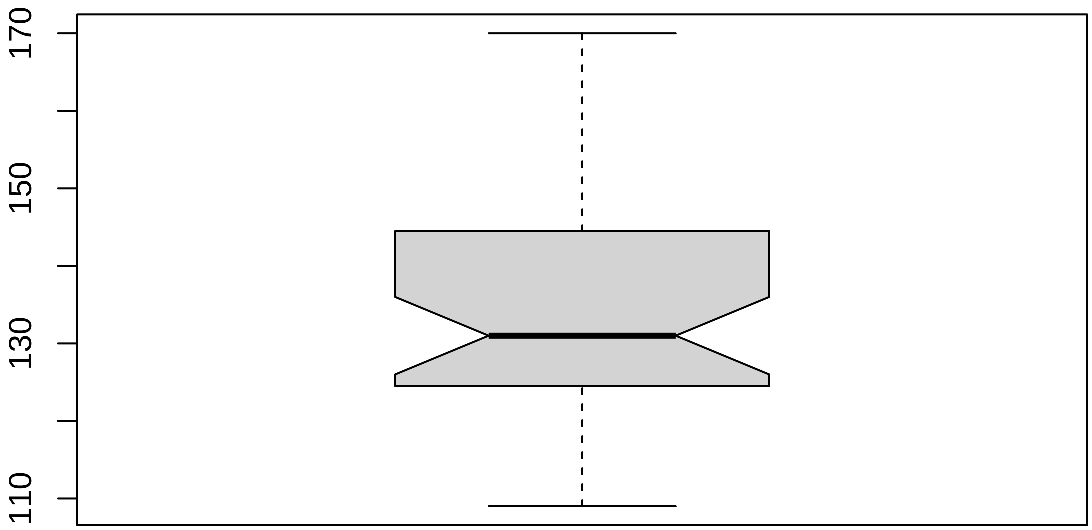
```
summary(castings$weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   109.0   124.8   131.0   134.8   143.8   170.0
```

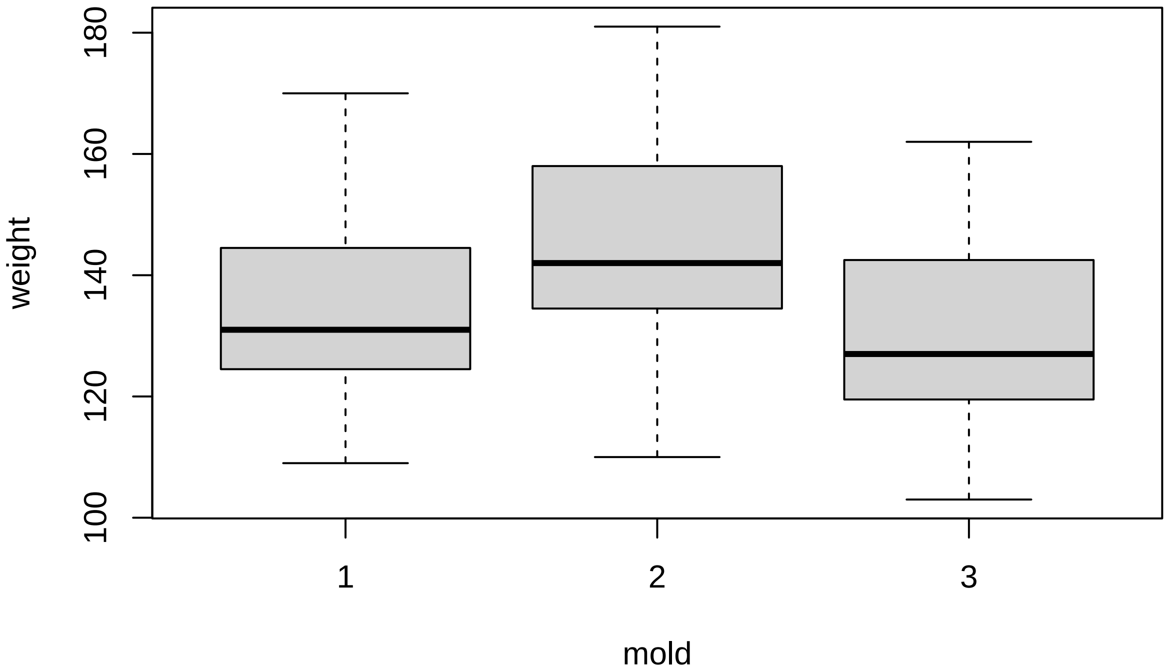
```
# Regular Boxplot
boxplot(castings$weight)
```




```
#Notched Box and Whiskers Plot
boxplot(castings$weight, notch=T)
```



```r
#Compare Groups
castings3 = read.csv("~/Documents/GitHub/school_cu/school_cu/methods for quality improvement/DTSA5704_Do

boxplot(weight ~ mold, data=castings3)
```



Measures of Central Tendency

The Mean

```r
weight = c(65, 67, 36, 37, 36, 57, 53, 39, 39, 58)

perform = data.frame(weight)
```

Calculations

Ungrouped Data : $\bar{X} = \frac{\sum X}{n}$

Grouped Data: $\bar{X} = \frac{\sum fX_c}{n}$ - X_{c} is the midpoint of each class interval - f is the frequency associated with each class interval

Weighted Mean: $\bar{X} = \frac{\sum w_j X}{w_j n_j}$

Weighted Mean can also be:

```r
#Calculate Mean
mean(perform$weight)
```

```
## [1] 48.7
```

```r
#Grouped Mean
fdcast = frequency.dist.grouped(castings$weight)

#Parentheses helps us see output
#Grab the midpt and freq from the grouped dataframe
midpts = fdcast$midpoint
freq = fdcast$freq

weighted.mean(x=midpts, w=freq)
```

```
## [1] 135.25
```

```r
#Weighted Mean
wt = c(0.2, 0.4, 0.4)
x = c(88, 85, 92)
weighted.mean(x=x, w=wt)
```

```
## [1] 88.4
```

Median and Mode

Median

```r
#Median
median(perform$weight)
```

```
## [1] 46
```

Mode

represented by M_o

Advantages - Not affected by extreme values

Disadvantage - The data set may not have a modal value - The data set may contain too many model values to be useful

```r
sample.mode(perform$weight)
```

```
## [1] 36 39
```

Measures of Position

Display values representing position or order in the data set or distribution. Examples are: - Low and High (X_L), (X_H) - Percentiles - Quartiles

```
min(perform)
```

Min_Max

```
## [1] 36
```

```
max(perform)
```

```
## [1] 67
```

Percentiles

- The Pth percentil is the value that P% of the values fall at or below and (100 - P%) fall above it
- Symbols: no common symbols used, but generally written simply as "Pth" percentile

How to calculate - First, sort the data from low to high - The Pth percentile is found int eh $\frac{1+P}{100^{th}}$ position (P in a proportion). - Example, find the 30th percentile with 1 + 0.3(n-1)th or 1 + 0.3(10-1) = 3.7 position.

```
quantile(x=perform$weight, probs=0.3)
```

```
##  30%
## 38.4
```

Quartiles are the 25th, 50th, 75th and 100th percentiles.

Measures of Dispersion and Shape

Measures of Dispersion reflect the variation of the spread in a data set or distribution. Some of the common measures of dispersion are: - Range - Interquartile Range - Semi-Interquartile Range - Standard Deviation - Variance

Range

- Advantages
  - Depends on only two values
  - Easy to understand
- Disadvantages
  - Extremely sensitive to outliers

```
rng = range(perform$weight)
rng[2] - rng[1]
```

```
## [1] 31
```

Interquartile Range

Range of the middle 50% of the distribution

```
IQR(perform$weight)
```

```
## [1] 20.25
```

Standard Deviation

Measure of variation that includes all data values in its calculations.

$$s = \sqrt{\frac{\sum(X - \bar{X})^2}{n-1}}$$

```
sd(perform$weight)
```

```
## [1] 12.57025
```

The Variance

The average squared distance values fall from the mean

$$s^2 = \frac{\sum(X - \bar{X})^2}{n-1}$$

```
var(perform$weight)
```

```
## [1] 158.0111
```

Measures of Shape

Skewness

- Concerned with the symmetrical nature of the distribution
- The degree of departure from symmetry of a distribution
- Symmetric distributions have a measure of 0
- Symbols (g_{3})
- The most important group of measures of skewness and kurtosis use the third and fourth moments of the mean
- Moments about the means are the average of the deviations from the mean raised to some power
- The sign displays the direction of skewness

```
#
round(skewness(castings$weight), 3)
```

```
## [1] 0.643
```

Kurtosis

Concerned with the peakedness of the distribution. The degree of peakedness of a distribution.

Different types: - Intermediate distribution with zero kurtosis is known as **mesokurtic** - A symmetrical **platykurtic** distribution has a lower peak and lighter tails, and has negative kurtosis - **Leptokurtic** distributions have heaveier tails and a taller peak - Sample: g_4

```
kurtosis(castings$weight)
```

```
## [1] -0.1690814
```

```
#Using the summary.continuous function
summary.continuous(castings$weight, stat.sd=T)
```

```
##   dv.name  n missing   mean      var       sd g3.skewness   g3test.p
## 1      fx 40       0 134.75 217.4744 14.74701   0.6431397 0.08469753
##   g4.kurtosis  g4test.p
## 1  -0.1690814 0.9691563
```

Measures of Relationship

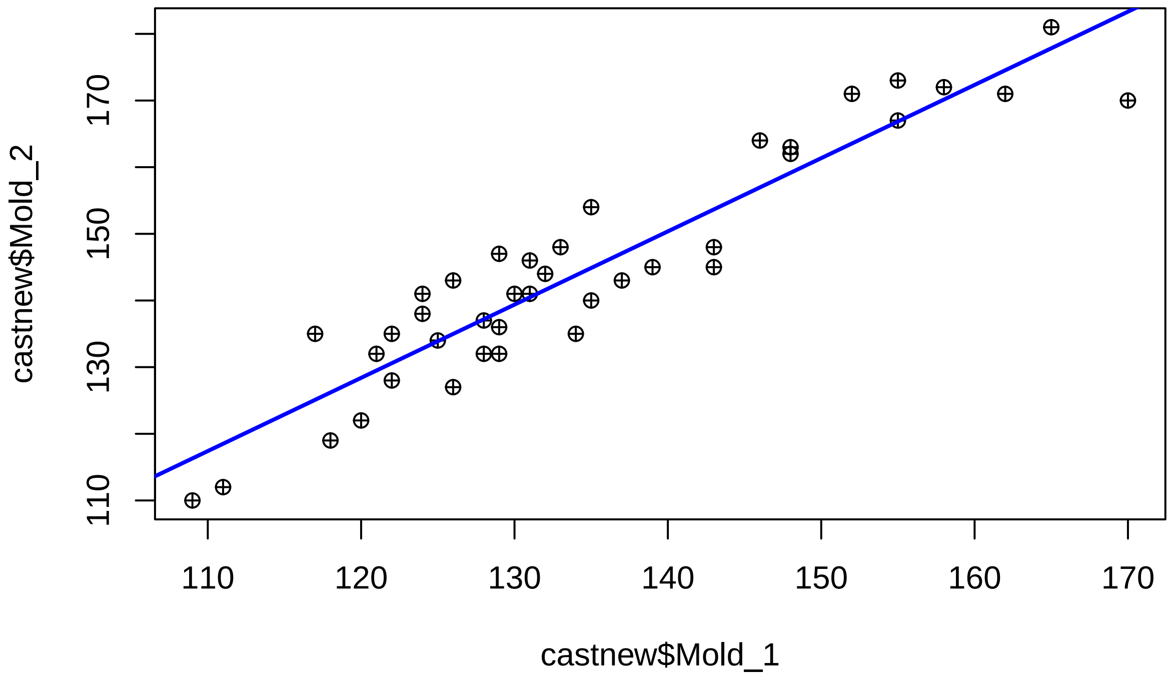
Correlation and association are measures of the strength of a relationship between two variables.

Difference between Correlation and Association Studying the relationship between two continuous variables is *correlation* while studying the relationshup between two nominal variables is *association*.

```
#Transform castings3 data from independent to dependent format
castnew = transform.independent.format.to.dependent.format(
  fx= weight~mold, data= castings3 )

#rename column headings
colnames(castnew)[1:3] = c("Mold_1", "Mold_2", "Mold_3")

#Calculate Correlation
cor(x=castnew$Mold_1, y=castnew$Mold_2, method="pearson")
```

```
## [1] 0.9363887
```

```
#Create Scatterplot
plot(x=castnew$Mold_1, y=castnew$Mold_2, pch=10, cex=1)
abline(lm(castnew$Mold_2 ~ castnew$Mold_1), col="blue", lwd=2)
```


castnew$Mold_2
110
130
150
170
110
120
130
140
150
160
170
castnew$Mold_1