MATH 3050 – Predictive Analytics

UNC CHARLOTTE

Topic 3: Classical Tests

- Tests of Normality
- Student t Distribution
- Bootstrapping
- Type I, II, III Error
- Contingency Tables
- K-S Test

1

Objectives of this Lesson:

By the end of this lesson you should be able to:

- Determine is a distribution is normally distributed
- Create residual analyses and probability plots
- Identify differences between a Student t distribution and the Normal distribution
- How to create a bootstrapped sample
- Define and calculate Type I, II, and III errors
- How to create contingency tables
- Test for significance of correlations
- Understand and calculate the Kolmogorov-Smirnov test of normality

2

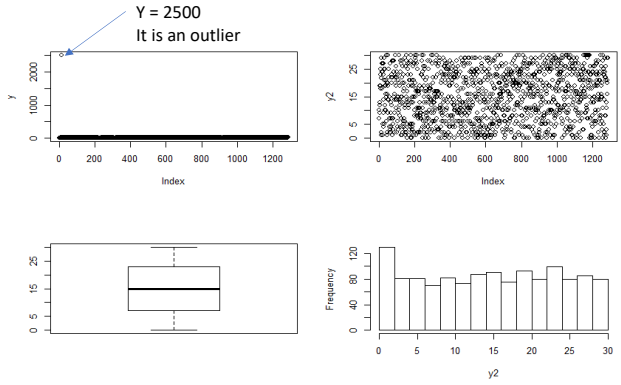
Some preliminary facts to establish regarding a data set:

1. Are the values normally distributed or not?
2. Are there outliers in the data?
3. If data were collected over a period of time, is there evidence for serial correlation?
4. Is the data skewed?
5. How highly correlated are the variables?
6. Is there missing data?

A yes to any of these questions could invalidate your inferences. This is a sampling of questions. This list is far from complete.

Summarizing Data

```
data <- read.table("c:\\temp\\classics2.txt",header=T)
names(data)
attach(data)
par(mfrow=c(2,2))

plot(y)

#Note y[11]= 2500
y[11] <- 21.75
plot(y)
boxplot(y)
hist(y,main="")
```



Topic 6: Classical Tests

Initial Data Analysis

```
summary(y)
```

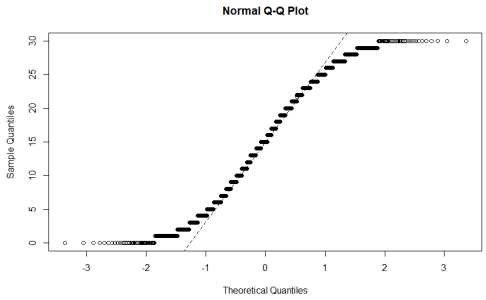
```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.00    7.00   15.00   15.13   23.00   30.00
```

```
fivenum(y)
```

```
[1]  0  7 15 23 30
```

Both functions give you Tukey's Five Famous Numbers. First also gives you the median.

```
#To plot the QQ Plot
par(mfrow=c(1,1))
qqnorm(y)
qqline(y,lty=2)
```



5

Plots for testing normality: Q-Q Plot ★★★★★

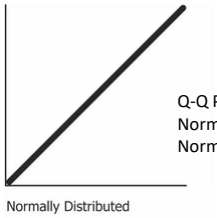
The simplest test of normality (and in many ways the best) is the Quantile–Quantile plot (Q-Q Plot).

This plots the ranked quantiles from our data sample against a similar number of ranked quantiles taken from a normal distribution. If our sample is normally distributed, then the line will be straight.

Q-Q Plots are measured against the Standard Normal Distribution.

Notice: When we plot the standard normal against itself, we get a straight line. In fact whenever you plot a data set against itself you get a straight line.

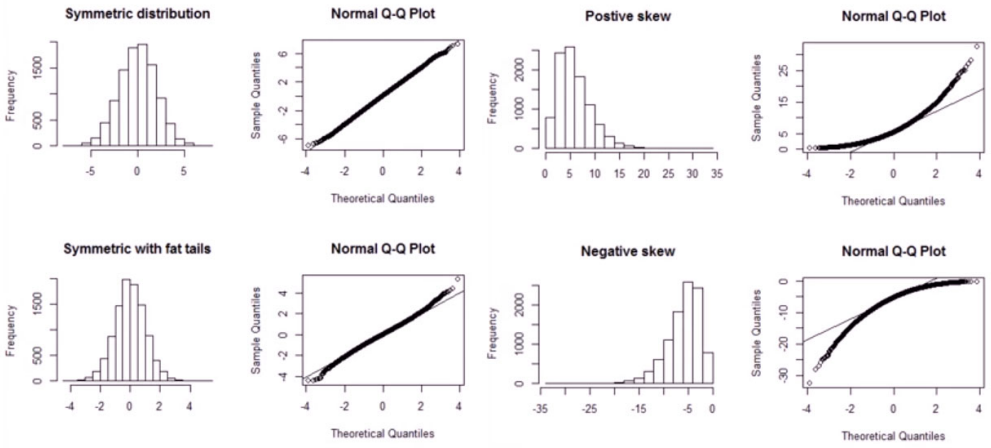
Try
y = sample(1000)
plot(y, y)



Q-Q Plot of a Standard Normal Against a Standard Normal

6

When you plot a non-standard normal against a standard normal, you don't get a straight line. The way the points depart from the straight line gives us some information about the shape of the given data set. ★★★★★



7

An Example Calculation

Do the following values come from a normal distribution?
7.19, 6.31, 5.89, 4.5, 3.77, 4.25, 5.19, 5.79, 6.79.

Step 1: **Order the items from smallest to largest**.

- 3.77
- 4.25
- 4.50
- 5.19
- 5.89
- 5.79
- 6.31
- 6.79
- 7.19

Let y_values <- c(3.11, 4.25, 4.50, 5.19, 5.79, 5.89, 6.31, 6.79, 7.19)

8

An Example Calculation

Step 2: **Draw a normal distribution curve. Divide the curve into n+1 segments. We have 9 values, so divide the curve into 10 equally-sized areas. For this example, each segment is 10% of the area (because 100% / 10 = 10%)**

- 10%
- 20%
- 30%
- 40%
- 50%
- 60%
- 70%
- 80%
- 90%
- 100%

9

An Example Calculation

Step 3: **Find the z-value (cut-off point) for each segment** in Step 3. These segments are *areas*, so refer to a z-table (or use software) to get a z-value for each segment.

The z-values are:

- 10% = -1.28
- 20% = -0.84
- 30% = -0.52
- 40% = -0.25
- 50% = 0
- 60% = 0.25
- 70% = 0.52
- 80% = 0.84
- 90% = 1.28
- 100% = 3.0

You can use qnorm(x, mean = 0, sd = 1) to verify all these values

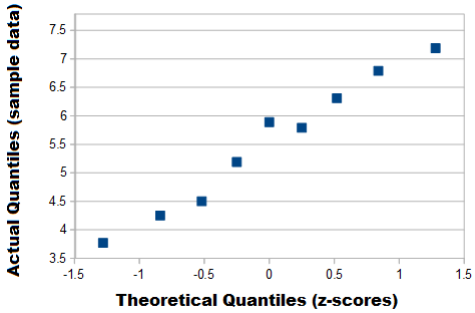
Let z_scores <- c(-1.28, -0.84, -052, -0.25, 0, 0.25, 0.52, 0.84, 1.28)

10

An Example Calculation

Step 4: Plot your data set values (Step 1) against your normal distribution cut-off points (Step 3).

```
#The expected graph if normally distributed
plot (z_scores, z_scores)

#Test the actual data
plot (z_scores, y_values)
```



11

Homework

1. Review video at https://www.youtube.com/watch?v=Erze9pNIX8A
2. Review video at https://www.youtube.com/watch?v=9IcaQwQkE9I
3. Write an R program to create the Q-Q Plot for the data set below. You do not have to recreate the theme.

```
Mydata <- c(15, 10, 25, 37, 42, 12, 40, 38, 50, 44)
```

Hints:
1. You will need to use the sort() function
2. You will need the abline() function to draw a smooth line through your points.

12

A Final Note ★★★★★

The normal Q-Q plot is one way to assess normality. However, you don't have to use the normal distribution as a comparison for your data; you can use any continuous distribution as a comparison (for example a Weibull distribution or a uniform distribution), as long as you can calculate the quantiles.

In fact, a common procedure is to test out several different distributions with the Q-Q plot to see if one fits your data well. Also, we would have to come up with different standards for skewness and kurtosis for those distributions.

Testing for Normality: The Shapiro-Wilk Test

The Shapiro-Wilk test is a way to tell if a random sample comes from a normal distribution. The test gives you a W value; small values indicate your sample is *not* normally distributed (you can reject the null hypothesis that your population is normally distributed if your values are under a certain threshold). The formula for the W value is:

$$W = \frac{\left(\sum_{i=1}^{n} a_i x_{(i)}\right)^2}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

where:
x_i are the ordered random sample values
a_i are constants generated from the covariances, variances and means of the sample (size n) from a normally distributed sample.

***** The test has limitations, most importantly the test has a bias by sample size. The larger the sample, the more likely you'll get a statistically significant result.

Testing for Normality: The Shapiro-Wilk Test ★★★★★

H_o: Data is Normally Distributed
H_1: Data is NOT Normally Distributed

```
x <- exp(rnorm(30))
shapiro.test(x)
```

```
      Shapiro-Wilk normality test
data: x
W = 0.5701, p-value = 3.215e-08
```


A Note About P-Values ★★★★★

A *p* value is ***not*** the probability that the null hypothesis is true (this is a common misunderstanding). On the contrary, the *p* value is based on the assumption that the null hypothesis *is* true.

A *p* value is an estimate of the probability that a particular result, or a result more extreme than the result observed, could have occurred by chance, *if the null hypothesis were true*.

In short, the *p* value is a measure of the credibility of the null hypothesis. A large *p* value (say, $p = 0.23$) means that there is no compelling evidence on which to reject the null hypothesis.

A Note About P-Values ★★★★★

Saying 'we do not reject the null hypothesis' and 'the null hypothesis is true' are two quite different things.

For instance, we may have **failed to reject a false null hypothesis** because our sample size was too low, or because our measurement error was too large.

Thus, *p* values are interesting, but they do not tell the whole story: **effect sizes** and **sample sizes** are equally important in drawing conclusions.

Type I Error

Null Hypothesis H_0: **Not Pregnant**

Type 1 Error

- The error of rejecting a hypothesis when it is true
- Called a False Positive
- Denoted by α

Type II Error

Null Hypothesis H_0: **Not Pregnant**

Type 2 Error

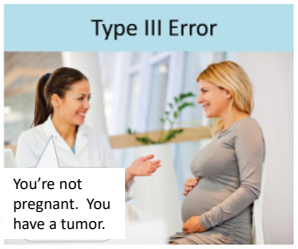
- The error of failing to reject a hypothesis when it is false
- Called a False Negative
- Denoted by β

Type III Error

Null Hypothesis H_0: **Not Pregnant**

Type 3 Error

- The error of rejecting a hypothesis for the wrong reason.



HW

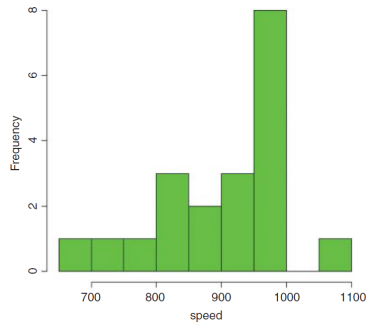
1. (T/F) A Shapiro-Wilk test that has a p-value of 0.03 would indicate that the variable of interest is not normally distributed.
2. For each of the following distributions, generate 1,000 random numbers and use the Shapiro test to test them for normality. State your conclusion for the results for each test based on the p-value.
    1. Beta (a=2, b=3)
    2. Weibull($\alpha = 2, \lambda = 4$)
    3. Logistic ($\mu=3, s = 2$)

The Wilcoxon Signed Rank Test ★★★★★

Wilcoxon signed rank test, a rank test used in nonparametric statistics, can be considered a t-test where dependent variable is not normally distributed.

Let's examine data from Michelson's famous experiment in 1879 to measure the speed of light.

```
light <- read.table("t:\\data\\light.txt",header=T)
attach(light)
hist(speed, main="",col="green")
```



The Wilcoxon Signed Rank Test

```
summary(speed)
```

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 650     850     940     909     980    1070
```

Observations

- The median is much larger than the mean. The first indication of skewness.
- If the median is larger than the mean, there is negative skewness.
- If the median is smaller than the mean, there is positive skewness.
- Rule of Thumb for Outliers: 1.5 times IQR below/above 1st / 3rd quantile: (655, 1175)
- There are no large outliers but one small one (14, 650). The 14th observation is 650.
  Note: 14 is the observation number. 650 is considered the outlier.

Light Data

	speed
14	650
2	740
15	760
16	810
1	850
6	850
10	880
3	900
5	930
13	930
7	950
19	960
20	960
8	980
9	980

The Wilcoxon Signed Rank Test

We want to test the hypothesis that Michelson's estimate of the speed of light is significantly different from the value of 299,990 thought to prevail at the time. Since the data have all had 299,000 subtracted from them, the test value is 990.

Because of the non-normality of the data, the use of the Student's *t* test in this case is ill advised. The correct test is Wilcoxon's signed-rank test.

```
wilcox.test(speed ,mu=990)
```

```
        Wilcoxon signed rank test with continuity correction

data: speed
V = 22.5, p-value = 0.00213
alternative hypothesis: true location is not equal to 990

Warning message:
In wilcox.test.default(speed, mu = 990) :cannot compute exact p-value with ties
```

We reject the null hypothesis and accept the alternative hypothesis because p = 0.00213 (i.e. much less than 0.05). The speed of light is significantly less than 299,990 given the distribution.

Bootstrap in Hypothesis Testing

You have probably heard the old phrase about 'pulling yourself up by your own bootlaces'. That is where the term 'bootstrap' comes from. It is used in the sense of getting 'something for nothing'.

The idea is very simple. You have a single sample of *n* measurements, but you can sample from this in very many ways, so long as you allow some values to appear more than once. This is called *sampling with replacement*.

25

Bootstrap in Hypothesis Testing

All you do is calculate the sample mean lots of times, once for each sampling from your data, then obtain the confidence interval by looking at the extreme highs and lows of the estimated means using a quantile function to extract the interval you want (e.g. a 95% interval is specified using `c(0.0275,0.975)` to locate the lower and upper bounds).

26

Bootstrap in Hypothesis Testing

Our sample mean value of the light data is 909.

How likely is it that the population mean that we are trying to estimate with our random sample of 100 values is as big as 990?

We take 10,000 random samples with replacement using n = 20 from the 20 values of light and calculate 10,000 values of the mean.

Then we ask: What is the probability of obtaining a mean as large as 990 by inspecting the right-hand tail of the cumulative probability distribution of our 10,000 bootstrapped mean values?

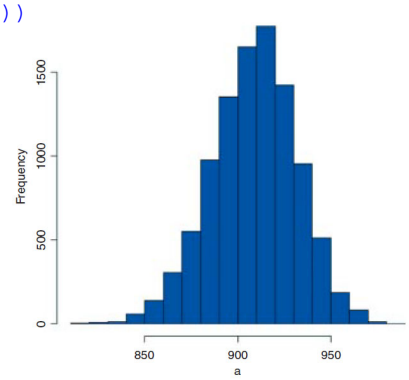
Bootstrap in Hypothesis Testing

```
light <- read.table("t:\\data\\light.txt",header=T)#Use your directory structure
attach(light)
a <- numeric(10000)
for(i in 1:10000) a[i] <- mean(sample(speed,replace=T))
hist(a,main="",col="blue")
```

This is the variable name in the light datafile.

The test value of 990 is way off the scale to the right, so a mean of 990 is clearly most unlikely, given the data with `max(a) = 979`.

In our 10,000 samples of the data, we never obtained a mean value greater than 979, so the probability that the mean is 990 is clearly $p < 0.0001$.



Format of a Typical Bootstrapping Routine ★★★★★

```
a <- numeric(10000)

for(i in 1:10000) a[i] <- mean(sample(speed,replace=T))

hist(a,main="",col="blue")
```

Calculates 10,000 means by sampling the variable speed and stores the results in a[i]

This technique is used a lot in statistics.

Skewness & Kurtosis ★★★★★

Skew (or skewness) is the dimensionless version of the third moment about the mean. It measures the extent to which a distribution has long, drawn-out *tails* on one side or the other.

$$m_3 = \frac{\sum (y - \bar{y})^3}{n}$$

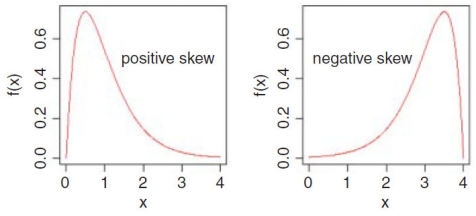
$$s_3 = \text{sd}(y)^3 = (\sqrt{s^2})^3$$

$$\text{skew} = \gamma_1 = \frac{m_3}{s_3}$$

Skewness & Kurtosis

A normal distribution is symmetrical and has $\gamma_1 = 0$. Negative values of γ_1 mean skew to the left (negative skew) and positive values mean skew to the right.

```
windows(7,4)
par(mfrow=c(1,2))
x <- seq(0,4,0.01)

plot(x,dgamma(x,2,2),type="l",ylab="f(x)",
xlab="x",col="red")
text(2.7,0.5,"positive skew")

plot(4-x, dgamma(x,2,2), type="l", ylab="f(x)",
xlab="x",col="red")
text(1.3,0.5,"negative skew")
```



Significance test for skewness

To test whether a particular value of skew is significantly different from 0 (and hence the distribution from which it was calculated is significantly non-normal) we divide the estimate of skew by its approximate standard error:

$$se_{\gamma_1} = \sqrt{\frac{6}{n}}$$

Function for Skewness ★★★★★

```
skew <- function(x){
m3 <- sum((x-mean(x))^3)/length(x)
s3 <- sqrt(var(x))^3
m3/s3
}
```

The last expression inside the function is not assigned a variable name and is returned as the value of `skew(x)` when this is executed from the command line.

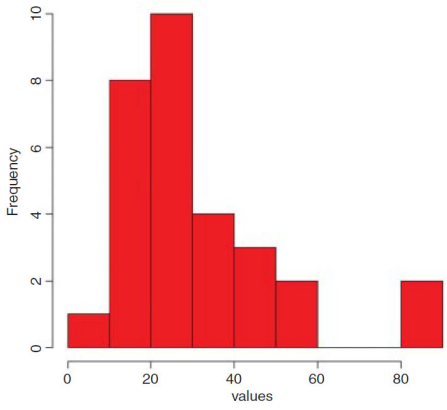
Example:

```
data <-read.table("c:\\temp\\skewdata.txt",header=T)
attach(data)
names(data)
hist(values)

skew(values)
skew(values)/sqrt(6/length(values))
```

What is the probability of getting a *t* value of 2.949 or larger by chance alone, when the skew value really is zero?

```
1-pt(2.949,29)
[1] 0.003121444
```

Cumulative Distribution Function for Student t



A Square Root Transformation of the Data

The goal of the transformation is to normalize the data to reduce the skewness.

```
skew(sqrt(values))/sqrt(6/length(values))
[1] 1.474851
```

This is not significantly skewed.

A Log- Transformation of the Data

The goal of the transformation is to normalize the data to reduce the skewness.

```
skew(log(values))/sqrt(6/length(values))
[1] -0.6600605
```

The distribution is now slightly skew to the left (negative skew), but the value of Student's *t* is smaller than with a square root transformation, so we might prefer a log transformation in this case.

Kurtosis ★★★★★

This is a measure of non-normality that has to do with the peakedness, or flat-toppedness, of a distribution. The normal distribution is bell-shaped, whereas a kurtotic distribution is other than bell-shaped. In particular, a more flat-topped distribution is said to be **platykurtic**, and a more pointy distribution is said to be **leptokurtic**.

The "−3" is included because a normal distribution has $m_4/s_4 = 3$. This formulation therefore has the desirable property of giving zero kurtosis for a normal distribution, while a flat-topped (platykurtic) distribution has a negative value of kurtosis, and a pointy (leptokurtic) distribution has a positive value of kurtosis.

The approximate standard error of kurtosis is $se_{\gamma_2} = \sqrt{\frac{24}{n}}$

$$m_4 = \frac{\sum (y - \bar{y})^4}{n}$$

$$s_4 = (\text{var}(y))^2 = (s^2)^2$$

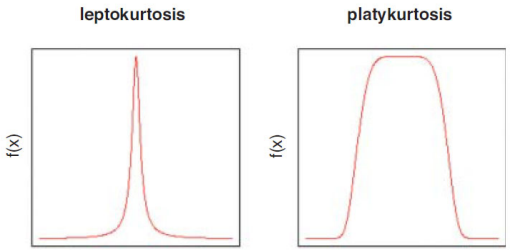
$$\text{kurtosis} = \gamma_2 = \frac{m_4}{s_4} - 3$$

Kurtosis Function

```
kurtosis <- function(x) {
m4 <- sum((x-mean(x))^4)/length(x)
s4 <- var(x)^2
m4/s4 - 3 }
```

For our present data, we find that kurtosis is not significantly different from normal:

```
kurtosis(values)
[1] 1.297751

kurtosis(values)/sqrt(24/length(values))
[1] 1.45093
```



Homework

Use the cat data set and write a program to do the following:

1. Read in the data set
2. Create exploratory plots and histograms for each level of the y variable
3. Create Q-Q plots for each level of the y variable and across all levels combined
4. What can you conclude from these plots?
5. Calculate the Shapiro-Wilks test for each level of the y variable and across all levels combined
6. Calculate the Wilcoxon Signed Rank Test for each level of the y variable and across all levels combined
7. Calculate skewness and kurtosis for each level of the y variable and across all levels combined
8. Calculate the probability of obtaining the skewness values in #7

Homework

The following code will create the dataset ("my_data")you need for this exercise:

```
set.seed(1234) #set.seed() will keep your random numbers from
         #changing each time you run the code

my_data <- data.frame(name = paste0(rep("M_", 10), 1:10),
       weight = round(rnorm(10, 20, 2), 1) )
```

Write a program that will do the following:

1. Print the first 10 rows of the data
2. Generate a summary table
3. Create a boxplot of the weight variable
4. Perform the Shapiro-Wilks Test on weight
5. Create a Q-Q plot
6. Perform a one-sample t-test on the weight variable and store the results in the variable "res" {Hint: t.test()}. Test H_0: mu = 25.
7. Print the p-value
8. Print the estimate #The estimate returned is the mean.
9. Print the confidence interval

Classic Tests for Two Samples

1. Comparing two variances (Fisher's *F* test, `var.test`);
2. Comparing two sample means with normal errors (Student's *t* test, `t.test`);
3. Comparing two means with non-normal errors (Wilcoxon's rank test, `wilcox.test`);
4. Comparing two proportions (the binomial test, `prop.test`);
5. Correlating two variables (Pearson's or Spearman's rank correlation, `cor.test`);
6. Testing for independence of two variables in a contingency table (chi-squared, `chisq.test`, or Fisher's exact test, `fisher.test`).

Degrees of Freedom

Degrees of freedom of an estimate from a sample is **the number of independent pieces of information that went into calculating the estimate**.

It's not quite the same as the number of items in the sample. In order to get the df for the estimate, you have to subtract 1 from the number of items and 1 for each parameter estimated by the model.

Let's say you were finding the mean weight loss for a low-carb diet. You could use 4 people, giving 3 degrees of freedom (4 – 1 = 3), or you could use one hundred people with df = 99. Both these examples assume no other parameters are estimated, only the mean.

Why Do Critical Values Decrease While DF Increase?

Let's take a look at the t-score formula in a hypothesis test:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

When n increases, the t-score goes up. This is because of the square root in the denominator: as it gets larger, the fraction s/√n gets smaller and the t-score (the result of another fraction) gets bigger.

As the degrees of freedom are defined above as n-1, you would think that the t-critical value should get bigger too, but they don't: they get *smaller*.

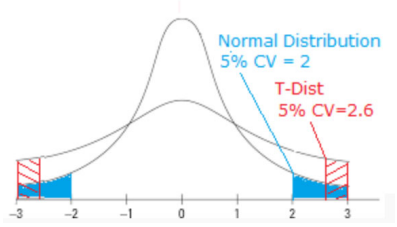
Why Use a t-test?

Use a t-test when the standard deviation of your population is unknown, which means you don't know the shape of it.

It could have short, fat tails. It could have long skinny tails. You just have no idea.

The degrees of freedom affect the shape of the graph in the t-distribution; as the df get larger, the area in the tails of the distribution get smaller.

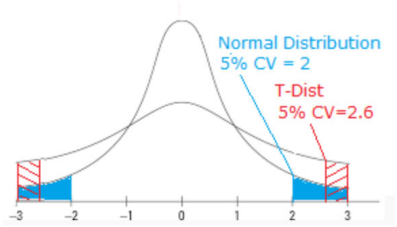
As df approaches infinity, the t-distribution will look like a standard normal distribution. When this happens, you can be certain of your standard deviation (which is 1 on a **standard** normal distribution)!



Why Use a t-test?

Additional Observations

- If you have a tiny sample say size = 4, the t-distribution will have fat tails. The fat tails tell you that you're more likely to have extreme values in your sample. You test your hypothesis at an alpha level of 5%, which cuts off the last 5% of your distribution.
- If the df increases, it also stands that the sample size is increasing; the graph of the t-distribution will have skinnier tails, pushing the critical value towards the mean. This observation address the last comment on slide 43.



Tests on Two Samples ★★★★★

Step 1: Test that the variances are not significantly different because this a required assumption of many statistical tests. If the variances are significantly different, no further testing should be done with a test that requires equal variances.

Step 2: Calculate test statistic for differences between means.

Step 3: Determine the appropriate number of degrees of freedom

Step 4: Obtain the *critical value* (a quantile) based on degrees of freedom and α level

Step 5: Compare critical value and test statistic

Step 6: Accept or reject the null hypothesis (H_0) based on the comparison

The Fisher *F* Test: Tests for Equal Variances ★★★★★

To compare two variances, all you do is divide the one variance by the other variance. Obviously, if the variances are the same, the ratio will be 1.

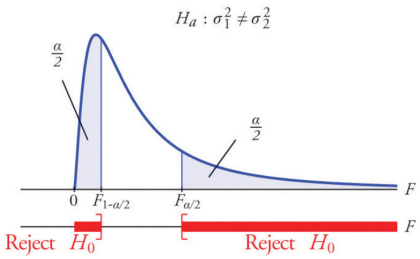
In order to be significantly different, the ratio will need to be significantly bigger than 1.

To test the ratio, we compare it to *critical value* of the variance ratio. The R function for this is qf(), which stands for 'quantiles of the *F* distribution'.

Because the F-Statistic is the ratio of two variances, the numerator will have a degrees of freedom and the denominator will have its owns degrees of freedom.

The Fisher *F* Test ★★★★★

Example: For our example of ozone levels in market gardens (see p. 354) there were 10 replicates in each garden, so there were 10 − 1 = 9 degrees of freedom for each garden. There are two gardens. The numerator and denominator each have 9 degrees of freedom. This is a two-tailed test where we split the α level between the two tails. (α = 0.05)




```
>qf(0.975,9,9)
[1] 4.025994
```

This means that a calculated variance ratio will need to be greater than or equal to 4.03 in order for us to conclude that the two variances are significantly different at $\alpha = 0.05$.

Example: Compare the variances in ozone concentration for market gardens B and C

```
f.test.data <- read.table("c:\\temp\\f.test.data.txt",header = T)
attach(f.test.data)
names(f.test.data)

var(gardenB)
[1] 1.333333

var(gardenC)
[1] 14.22222

criticalvalue <-qf(0.975,9,9)
criticalvalue
[1] 4.025994

F.ratio <- var(gardenC)/var(gardenB)
F.ratio
[1] 10.66667
```

Conclusion: The test statistic is greater than the critical value, therefore, we reject the null hypothesis and conclude the variances are significantly different.

The probability of getting an F Statistic as large as 10.67 or larger is given by

```
2*(1-pf(F.ratio,9,9))
[1] 0.001624199
```

Should we proceed with the t-test on the means? Why or Why Not?

Alternate formula

```
var.test(gardenC,gardenB)

        F test to compare two variances

data:  gardenC and gardenB
F = 10.667, num df = 9, denom df = 9, p-value = 0.001624
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  2.649449 42.943938
sample estimates:
ratio of variances
          10.66667
```

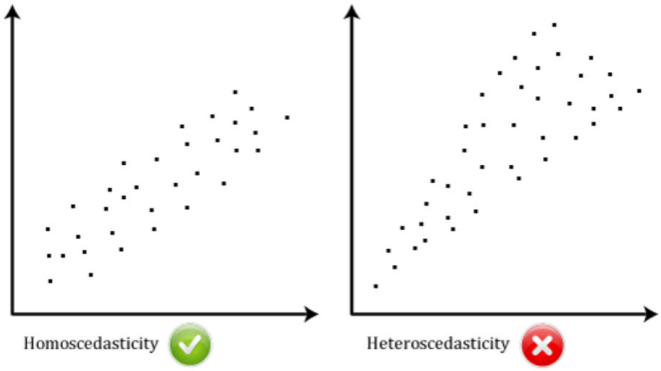
Note: You can also extract elements from the output

```
vartest = var.test(gardenB,gardenC)
names(vartest)
vartest$statistic
```

Homoscedasticity v. Heteroscedasticity ★★★★★

Definitions:
Homoscedasticity – Equal Scatter
Heteroscedasticity – Unequal Scatter

Constancy of variance (**homoscedasticity**) is the most important assumption underlying linear regression and analysis of variance.



Comparing Variance Across Multiple Samples

For multiple samples you can choose between the Bartlett test and the Fligner–Killeen test.
We use the refuge data for these tests.

```
refs <- read.table("c:\\temp\\refuge.txt",header=T)
attach(refs)
names(refs)
[1] "B" "T"
```

where *T* is an ordered factor with nine levels. Each level produces 30 estimates of yields except for level 9 which is a single zero.

```
tapply(B,T,var)
       1        2        3        4        5        6        7        8 9
1354.024 2025.431 3125.292 1077.030 2542.599 2221.982 1445.490 1459.955 NA
```

```
which(T==9)
[1] 31
```

We need to ignore this observation in calc because of NA value.

Bartlett's test for Homogeneity of Variances ★★★★★

It is used to test that variances are equal for all samples. It checks that the assumption of equal variances is true before running certain statistical tests like the One-Way ANOVA. It's used when you're fairly certain your data comes from a normal distribution. A similar test, called **Levene's test**, is a better choice for non normal distributions.

The null hypothesis for the test is that the variances are equal for all samples. In statistics terms, that's:
H_0: $\sigma_1^2 = \sigma_2^2 = \ldots = \sigma_k^2$.

The alternate hypothesis (the one you're testing), is that the variances are not equal for one pair or more:
H_0: $\sigma_1^2 \neq \sigma_2^2 \neq \ldots \neq \sigma_k^2$.

$$\chi^2 = \frac{(N-k)\ln(S_p^2) - \sum_{i=1}^{k}(n_i - 1)\ln(S_i^2)}{1 + \frac{1}{3(k-1)}\left(\sum_{i=1}^{k}\left(\frac{1}{n_i - 1}\right) - \frac{1}{N-k}\right)}$$

Fligner–Killeen Test ★★★★★

The Fligner Killeen test is a non-parametric test for homogeneity of group variances based on ranks. It is useful when the data is non-normal or where there are outliers.

$$FK = \frac{\sum_{j=1}^{k} n_j(\bar{a}_j - \bar{a})^2}{s^2}$$

where k = the number of groups, n_j = the size of the jth group, a_j-bar is the mean of the normalization values for the jth group, a-bar is the mean of all the normalization values and s^2 is the variance of all the normalization values.

Example:

```
refs <- read.table("c:\\temp\\refuge.txt",header=T)
attach(refs)
names(refs)
tapply(B,T,var)

which(T==9)

bartlett.test(B[-31],T[-31])
fligner.test(B[-31],T[-31])
```


The Results

```
bartlett.test(B[-31],T[-31])
       Bartlett test of homogeneity of variances
data: B[-31] and T[-31]
Bartlett's K-squared = 13.1986, df = 7, p-value = 0.06741
```

So there is no significant difference between the eight variances ($p = 0.067$). Now Fligner:

```
fligner.test(B[-31],T[-31])
       Fligner-Killeen test of homogeneity of variances
data: B[-31] and T[-31]
Fligner-Killeen:med chi-squared = 14.3863, df = 7, p-value = 0.04472
```

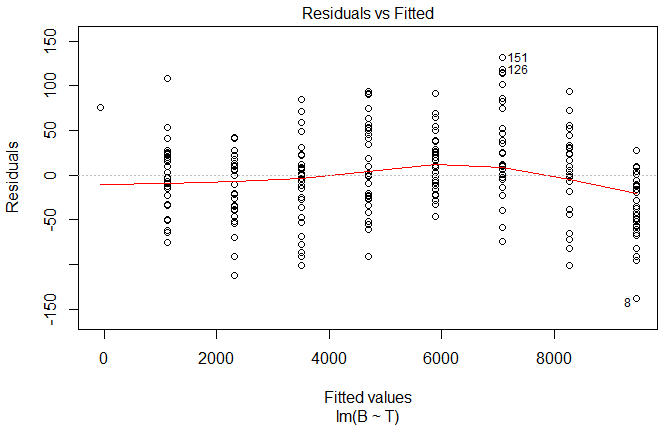
Hmm. This test says that there *are* significant differences between the variances ($p < 0.05$).

What should we do now?

Let's create some visualizations

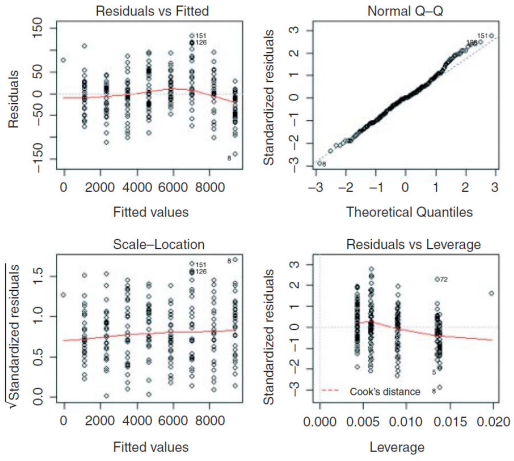
```
model <- lm(B~T)
plot(model)
```

Note: The fitted line is fairly close to the horizontal line suggesting normality.



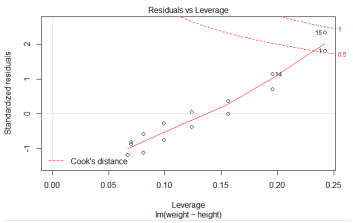
Analysis of Residuals

- The residual versus the fitted plot does not seem to have a strong nonlinear pattern. The line is fairly horizonal.
- The Q-Q Plot shows nonlinearity arounds the ends.
- The square root of the standardized residuals called a spread location plot depicts residuals that are random with no recognizable pattern.
- The leverage graph attempts to finds outliers which would over influence the orientation of the regression line.

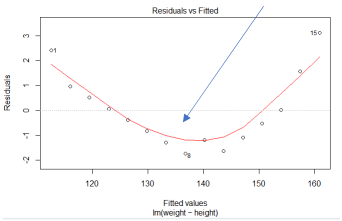


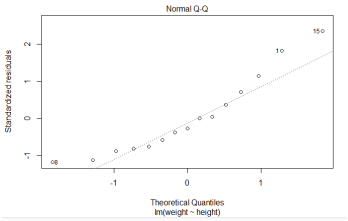
Plotting Fit Visualizations ★★★★★

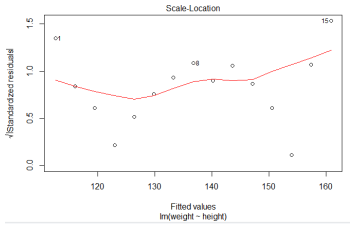
```
data<- women # Load a built-in data called 'women'
fit = lm(weight ~ height, women) # Run a regression analysis
plot(fit)  #Plot the argument fit will give 4 diagnostic graphs
```



Shape suggests a transformation is necessary to coerce a more linear fit.





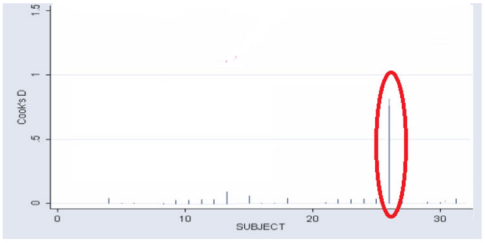


Cook's Distance (d)

In statistics, Cook's distance or Cook's D is a commonly used estimate of the influence of a data point when performing a least-squares regression analysis.

Technically, Cook's D is calculated by removing the i^{th} data point from the model and recalculating the regression. It summarizes how much all the values in the regression model change when the i^{th} observation is removed. The formula for Cook's distance is:

$$D_i = \frac{\sum_{j=1}^{n}(\hat{Y}_j - \hat{Y}_{j(i)})^2}{(p+1)\,\hat{\sigma}^2}$$



Cook's distance showing item #26 as a potential outlier.

Cook's Distance (d)

Several interpretations for Cook's distance exist.

- A **general rule of thumb** is that observations with a Cook's D of more than 3 times the mean, μ, is a possible outlier.
- An alternative interpretation is to investigate any point with a D_i over 4/n, where n is the number of observations.
- Other authors suggest that any "large" D_i should be investigated. How large is "too large"? The consensus seems to be that a D_i value of more that 1 indicates an influential value, but you may want to look at values above 0.5. Any value that sticks out from the other (like the one in the above chart) should also be investigated.
- An alternative (but slightly more technical) way to interpret D_i is to find the potential outlier's percentile value using the F-distribution. A percentile of over 50 indicates a highly influential point.

61

Example

```
ozone <- read.table("c:\\temp\\gardens.txt",header=T)
attach(ozone)
names(ozone)

y <- c(gardenA,gardenB,gardenC)
garden <- factor(rep(c("A","B","C"),c(10,10,10)))
```

The question is whether the variance in ozone concentration differs from garden to garden or not. Fisher's *F* test comparing gardens B and C says that variance is significantly greater in garden C:

```
var.test(gardenB,gardenC)
bartlett.test(y~garden)
fligner.test(y~garden)
```


62

The Results

```
var.test(gardenC,gardenB)

        F test to compare two variances

data:  gardenC and gardenB
F = 10.667, num df = 9, denom df = 9, p-value = 0.001624
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
  2.649449 42.943938
sample estimates:
ratio of variances
          10.66667
```


63

The Results

```
var.test(gardenB,gardenC)

        F test to compare two variances

data: gardenB and gardenC
F = 0.0938, num df = 9, denom df = 9, p-value = 0.001624
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
0.02328617 0.37743695
sample estimates:
ratio of variances
        0.09375
```


64

The Results

```
bartlett.test(y~garden)

        Bartlett test of homogeneity of variances
data: y by garden
Bartlett's K-squared = 16.7581, df = 2, p-value = 0.0002296
```

Bartlett's test says there is a highly significant difference in variance across gardens.

The Results

```
fligner.test(y~garden)

        Fligner-Killeen test of homogeneity of variances
data: y by garden
Fligner-Killeen: med chi-squared = 1.8061, df = 2, p-value = 0.4053
```

In contrast, the Fligner–Killeen test (preferred over Bartlett's test by many statisticians) says there is no compelling evidence for non-constancy of variance (**heteroscedasticity**) in these data

The reason for the difference is that Fisher and Bartlett are sensitive to outliers, whereas Fligner–Killeen is not (it is a non-parametric test which uses the ranks of the absolute values of the centered samples, and weights

a(i) = qnorm((1 + i/(n+1))/2).

Of the many tests for homogeneity of variances, this is the most robust against departures from normality

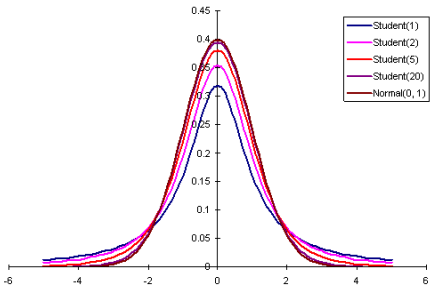
Two Classic Tests for Comparing Means ★★★★★

1. **Student's *t* test** when the samples are independent, the variances constant, and the errors are normally distributed;

2. **Wilcoxon's rank-sum test** when the samples are independent, but the errors are *not* normally distributed (e.g. they are ranks or scores).

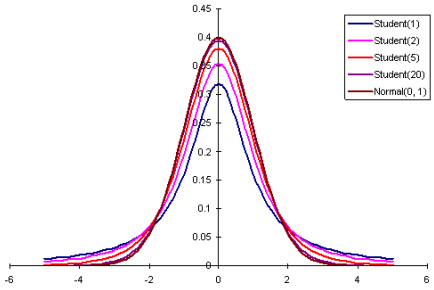
The Student t Distribution ★★★★★

The t-distribution (Student's t-distribution) is a continuous probability distribution that arises in estimating the mean of a normally distributed population when the sample size is small, and the population standard deviation is unknown. The larger the sample, the more the t-distribution resembles a normal distribution.



The Student t Distribution ★★★★★

dt(x, df, ncp, log = FALSE)
pt(q, df, ncp, lower.tail = TRUE, log.p = FALSE)
qt(p, df, ncp, lower.tail = TRUE, log.p = FALSE)
rt(n, df, ncp)



Arguments

x, q	vector of quantiles.
p	vector of probabilities.
n	number of observations. If length(n) > 1, the length is taken to be the number required.
df	degrees of freedom (> 0, maybe non-integer). df = Inf is allowed. For qt only values of at least one are currently supported.
ncp	non-centrality parameter delta; currently except for rt(), only for abs(ncp) <= 37.62. If omitted, use the central t distribution.
log, log.p	logical; if TRUE, probabilities p are given as log(p).
lower.tail	logical; if TRUE (default), probabilities are P[X <= x], otherwise, P[X > x]

The Student t Distribution

The t-test is commonly used to determine whether the means of two groups are equal to each other. The assumption for the test is that both groups are sampled from normal distributions with equal variances.

The null hypothesis (H_0) is that the two means are equal.
The alternative (H_1) is that they are not.

We can calculate a t-statistic that will follow a t-distribution with n1+ n2 - 2 degrees of freedom.

There is also a widely used modification of the t-test, known as Welch's t-test that adjusts the number of degrees of freedom when the variances are thought not to be equal to each other.

It is named for its creator, statistician Bernard Lewis Welch, and is an adaptation of Student's t-test.

The Student *t* test was developed by William Sealy Gosset, a chemist working for the Guinness brewery in Dublin, Ireland. "Student" was his pen name.

The Student t Distribution

Let's test it out on a simple example, using data simulated from a normal distribution.

```
x = rnorm(10)
y = rnorm(10)
t.test(x,y)
```

```
Welch Two Sample t-test

data:  x and y
t = 1.4896, df = 15.481, p-value = 0.1564
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.3221869  1.8310421
sample estimates:
 mean of x  mean of y
 0.1944866 -0.5599410
```


The Student t Distribution

It is useful to be able to extract metrics from the output

```
ttest = t.test(x,y)
names(ttest)
```

```
[1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
[6] "null.value"  "alternative" "method"      "data.name"
```

The value we want is named "statistic". To extract it, we can use the dollar sign notation, or double square brackets:

```
ttest$statistic
ttest[['statistic']]
```

Both statements return the value 1.489560

Student's *t* test

The test statistic is the number of standard errors of the difference by which the two-sample means are separated:

$$t = \frac{\text{difference between the two means}}{\text{standard error of the difference}} = \frac{\bar{y}_A - \bar{y}_B}{se_{\text{diff}}}$$

$$se_{\text{diff}} = \sqrt{\frac{s_A^2}{n_A} + \frac{s_B^2}{n_B}}$$

Square Root of Sum of Separate Variance

$$Df = \sum_{i=1}^{n} Df_i \; = Degree\ of\ Freedom\ for\ test\ statistic$$

Student's *t* test Example

H_0: Two population means are the same
H_1: Two population means are NOT the same

```
qt(0.975,18)
t.test.data <- read.table("c:\\temp\\t.test.data.txt",header=T)
attach(t.test.data)
par(mfrow=c(1,1))
names(t.test.data)
ozone <- c(gardenA,gardenB)
label <- factor(c(rep("A",10),rep("B",10)))
boxplot(ozone~label,notch=T, xlab="Garden",ylab="Ozone",col="red")
```

Student's *t* test Example

What is the significance of the non-overlappng notches?

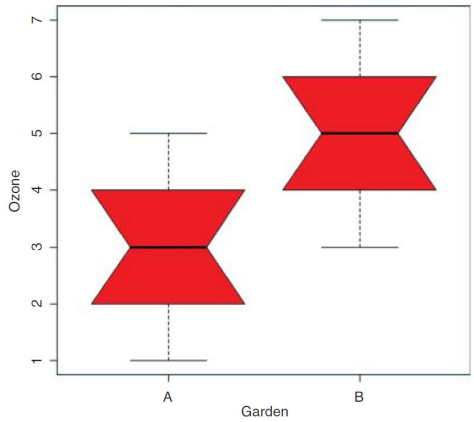
```
s2A <- var(gardenA)
s2B <- var(gardenB)

(mean(gardenA) -
      mean(gardenB))/sqrt(s2A/10+s2B/10)
```

[1] -3.872983 (Note the sign is irrelevant in a t test. Only Absolute Values are of Interest.)

```
2*pt(-3.872983,18) #Multiply by 2 as it is a two-tailed test
```

[1] 0.001114540

```
t.test(gardenA,gardenB)
```

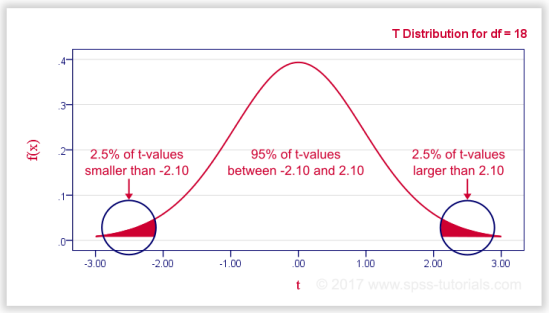


Remember: In a symmetric distribution the means and medians are the same.

75

Student's *t* test Example

```
t.test(gardenA,gardenB)

        Welch Two Sample t-test
data: gardenA and gardenB
t = -3.873, df = 18, p-value = 0.001115
alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:
-3.0849115 -0.9150885
sample estimates:
mean of x mean of y
        3         5
```



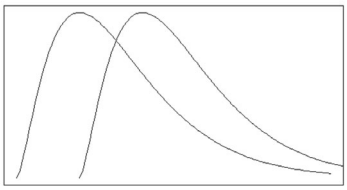
76

Wilcoxon Rank-Sum Test

This is a non-parametric alternative to Student's t test, which we could use if the errors were non-normal. This if for matched pairs data. The tests of hypotheses is:

H_0: The observations come from the same population
H_1: The observations DO NOT come from the same population

Identical distributions with different medians



From a practical point of view, this implies:

H_0: If one observation is made at random from each population (call them x_0 and y_0), then the probability that $x_0 < y_0$ is the same as the probability that $x_0 > y_0$, and so the populations for each sample have the **same medians**.

Wilcoxon Rank-Sum Test Using R Functions

```
ozone <- c(gardenA, gardenB)
label <- c(rep("A",10),rep("B",10))
combined.ranks <- rank(ozone)
tapply(combined.ranks, label, sum)
wilcox.test(gardenA,gardenB)
```

Note:
1. `rank()` handles tied ranks by averaging the appropriate ranks
2. `tapply` sums the ranks by label type, A (for Garden A) and B for Garden B

Wilcoxon Rank-Sum Test Using R Functions

```
wilcox.test(gardenA,gardenB)

        Wilcoxon rank sum test with continuity correction

data: gardenA and gardenB
W = 11, p-value = 0.002988
alternative hypothesis: true location shift is not equal to 0

Warning message:
In wilcox.test.default(gardenA, gardenB) :
cannot compute exact p-value with ties
```

There are tables of critical values associated with this test to determine p – values.

We will calculate W manually to confirm this value.

Correction for Continuity

When we use the normal approximation the phrase "with continuity correction" is added to the name of the test.

A continuity correction is an adjustment that is made when a discrete distribution is approximated by a continuous distribution. The normal approximation is very good and computationally faster for samples larger than 50.

Calculation of W in R

Recall that the Wilcoxon Sum Rank Test is a test of whether the medians of two distributions are shifted to the right or left of each other by answer the question:

"If one observation is made at random from each population (call them x_0 and y_0), then what is the probability that $x_0 < y_0$ or that $x_0 > y_0$?" What about when $x_0 = y_0$?

We can answer this question empirically:

R Calculation of Wilcoxon Sum Rank Test with Continuity Correction										
Garden A	3.0	4.0	4.0	3.0	2.0	3.0	1.0	3.0	5.0	2.0
Garden B	5.0	5.0	6.0	7.0	4.0	4.0	3.0	5.0	6.0	5.0
Question: How many times are there elements in Garden B less than each element in Garden A?										
Answer:	0.5	2	2	0.5	0	0.5	0	0.5	5	0
W =	11									

Homework

1. Use the file "scores.xls" to write a program to do the following:

   A. Exploratory plots including box plots of the relationship between the two groups
   B. What are your observations?
   C. Calculate Fligner–Killeen Test and report significant level in a single line of output
   D. Calculate the Student t and report significant level in a single line of output
   E. Calculate the Wilcoxon Signed Rank Test and report significant level in a single line of output

Homework

2. Use the file “drugtesting.csv” to write a program to do the following:

   A. Exploratory plots including box plots of the relationship between the two groups
   B. What are your observations?
   C. Calculate Fligner–Killeen Test and report significant level in a single line of output
   D. Calculate the Student t and report significant level in a single line of output
   E. Calculate the Wilcoxon Signed Rank Test and report significant level in a single line of output

Homework

3. Use following code to write the program as instructed below

```
# Data in two numeric vectors
women_weight <- c(38.9, 61.2, 73.3, 21.8, 63.4, 64.6, 48.4, 48.8, 48.5)
men_weight <- c(67.8, 60, 63.4, 76, 89.4, 73.3, 67.3, 61.3, 62.4) # Create a data frame
my_data <- data.frame( group = rep(c("Woman", "Man"), each = 9),
                       weight = c(women_weight, men_weight))
```

   A. Exploratory plots including box plots of the relationship between the two groups
   B. What are your observations?
   C. Calculate Fligner–Killeen Test and report significant level in a single line of output
   D. Calculate the Student t and report significant level in a single line of output
   E. Calculate the Wilcoxon Signed Rank Test and report significant level in a single line of output

Chi-squared contingency tables

A great deal of statistical information comes in the form of *counts* (whole numbers or integers): the number of animals that died, the number of branches on a tree, the number of days of frost, the number of companies that failed, the number of patients who died.

In statistics, however, the contingencies are *all the events that could possibly happen*.

A contingency table shows the counts of how many times each of the contingencies actually happened in a particular sample.

85

Chi-squared contingency tables

Example: Consider the following example that has to do with the relationship between hair color and eye color. We take a random sample and fill the table below.

	Blue eyes	Brown eyes
Fair hair	38	11
Dark hair	14	51

86

Chi-squared contingency tables

Next Step: We need a model to predict expected frequencies.

Assumption: Hair color is independent of eye color. This assumption it makes it possible to predict the expected frequencies based on the assumption that the model is true.

H_0: Hair color is independent of eye color.
H_1: Hair color is NOT independent of eye color.

	Blue eyes	Brown eyes	Row totals
Fair hair	38	11	49
Dark hair	14	51	65
Column totals	52	62	114

Probabilities of Contingency Table

What is the probability of getting a random individual from this sample whose hair was fair?

A total of 49 people (38 + 11) had fair hair out of a total sample of 114 people.
So the probability of fair hair is 49/114 and the probability of dark hair is 65/114.

What is the probability of selecting someone at random from this sample with blue eyes?

A total of 52 people had blue eyes (38 + 14) out of the sample of 114.
So the probability of blue eyes is 52/114 and the probability of brown eyes is 62/114.

Probabilities of Contingency Table

Now we can calculate the probabilities of having each of the two characteristics.

	Blue eyes	Brown eyes	Total count in each row
Fair hair	49/114 × 52/114	49/114 × 62/114	49
Dark hair	65/114 × 52/114	65/114 × 62/114	65
Total count in each column	52	62	114

It is important to note that the probabilities sum to 1.0.

The probability calculation simplifies to: $E = \frac{R\ x\ C}{G^2}$

Probabilities of Contingency Table

Now that we have the probabilities, we can calculate expected frequencies.
We simple multiply the cell probabilities by the total sample size.

	Blue eyes	Brown eyes	Row totals
Fair hair	22.35	26.65	49
Dark hair	29.65	35.35	65
Column totals	52	62	114

The frequency calculation simplifies to: $E = \frac{R\ x\ C}{G}$

Homework

Calculate the missing probabilities for the contingency table below.

	Data Scientist	Actuary	*Total*
Male			240
Female			100
Total	200	140	340

It is important to note that the probabilities sum to 1.0.

Homework

Calculate the missing expected frequencies for the contingency table below.

	Data Scientist	Actuary	*Total*
Male			240
Female			100
Total	200	140	340

It is important the frequencies sum to the grand total which is 340 in the above example.

The Null Hypothesis

We want to answer the question: Are the expected frequencies are *significantly* different from the observed frequencies?

We need a test statistic to address.

We examine two options:

1. Pearson's chi-squared
2. Fisher's exact test

Pearson's Chi-Squared

The test statistic $\chi 2$ is

$$\chi^2 = \sum \frac{(O - E)^2}{E}$$

where O is the observed frequency and E is the expected frequency.

Pearson's Chi-Squared

It makes the calculations easier if we write the observed and expected frequencies in parallel columns, so that we can work out the corrected squared differences more easily.

	O	E	$(O-E)^2$	$\frac{(O-E)^2}{E}$
Fair hair and blue eyes	38	22.35	244.92	10.96
Fair hair and brown eyes	11	26.65	244.92	9.19
Dark hair and blue eyes	14	29.65	244.92	8.26
Dark hair and brown eyes	51	35.35	244.92	6.93

Is this a big value of chi-squared or not?

$$\chi^2 = 35.33$$

Pearson's Chi-Squared

To work out the critical value of chi-squared we need two things:

1. The number of degrees of freedom, and
2. The degree of certainty with which to work.

A contingency table has a number of rows (r) and a number of columns (c), and the degrees of freedom is given by

$$\text{d.f.} = (r - 1) \times (c - 1).$$

It is conventional to say we want to be 95% certain about the falseness of the null hypothesis. This means $\alpha = 0.05$

A Note About Degrees of Freedom

	Blue eyes	Brown eyes	Row totals
Fair hair			49
Dark hair			65
Column totals	52	62	114

← Before you fill any of the boxes each has freedom to vary.

Once one box is filled, the rest can be determined. This means that for a 2 x 2 table there is only one degree of freedom. →

	Blue eyes	Brown eyes	Row totals
Fair hair		11	49
Dark hair			65
Column totals	52	62	114

Follow up

We have rejected the null hypothesis that eye color and hair color are independent. We have to established the *way* in which they are related (e.g. is the correlation between them positive or negative?).

To do this we need to look carefully at the data and compare the observed and expected frequencies.

If fair hair and blue eyes were positively correlated, would the observed frequency be greater or less than the expected frequency?

Ans: The observed frequency will be greater than the expected frequency when the traits are positively correlated (and less when they are negatively correlated).

In our case we expected only 22.35 but we observed 38 people (nearly twice as many) to have both fair hair and blue eyes. So, it is clear that fair hair and blue eyes are *positively* associated.

The R Procedure

```
count <- matrix(c(38,14,11,51),nrow=2)
count
```

```
     [,1] [,2]
[1,]   38   11
[2,]   14   51
```

Enter the data *columnwise* (not row-wise) into the matrix. Then the test uses the chisq.test function, with the matrix of counts as its only argument:

```
chisq.test(count)
```

```
        Pearson's Chi-squared test with Yates' continuity correction
data: count
X-squared = 33.112, df = 1, p-value = 8.7e-09
```


The R Procedure

The calculated value of chi-squared is slightly different from ours, because Yates' correction has been applied as the default. If you switch the correction off correct=F), you get the value we calculated by hand:

```
chisq.test(count,correct=F)
```

```
        Pearson's Chi-squared test
data: count
X-squared = 35.3338, df = 1, p-value = 2.778e-09
```

The R Procedure

If you need to extract the frequencies expected under the null hypothesis of independence, then use:

```
chisq.test(count,correct=F)$expected
```

```
     [,1]     [,2]
[1,] 22.35088 26.64912
[2,] 29.64912 35.35088
```


Homework: Chi-Square

1. (T/F) As the number of degrees of freedom increases, the graph of the chi-square distribution looks more and more symmetrical.
2. (T/F) The standard deviation of the chi-square distribution is twice the mean.
3. (T/F) In a goodness-of-fit test, the expected values are the values we would expect if the null hypothesis were true.
4. Which of the following statement is true regarding a Chi-Square Distribution:
   A. It is not symmetrical and is not highly skewed.
   B. It is symmetrical and is not highly skewed.
   C. It is not symmetrical and is highly skewed.
   D. It is symmetrical and is highly skewed.

5. Which of the following statement is true regarding a Chi-Square Distribution:
   A. Its expected value is n
   B. Its variance is 2n
   C. Categories can have expected frequencies < 1
   D. More than 25% of the categories can have frequencies < 5

Homework: Chi-Square

6. (T/F) The distribution of a Chi Square approaches a normal as n →∞
7. A six-sided die is rolled 120 times. Fill in the expected frequency column. Then, conduct a hypothesis test to determine if the die is fair. The data in Table are the result of the 120 rolls.

Face Value	Frequency	Expected Frequency
1	15	
2	29	
3	16	
4	15	
5	30	
6	15	

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

When one or more of the expected frequencies is less than 4 (or 5 depending on the rule of thumb you follow) then it is wrong to use Pearson's chi-squared for your contingency table.

This is because small expected values inflate the value of the test statistic, and it no longer can be assumed to follow the chi-squared distribution. The individual counts are *a, b, c* and *d* like this:

	Column 1	Column 2	Row totals
Row 1	a	b	$a + b$
Row 2	c	d	$c + d$
Column totals	$a + c$	$b + d$	n

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

Fisher's exact test is based on the hypergeometric distribution.

	A	Not A	*Column Total*
In Sample	a	b	*a+b*
Not In Sample	c	d	*c+d*
Row Total	*a+c*	*b+d*	*n*

This means we can ask the questions: What is the probability of obtaining a & c given the row, column and grand totals?

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

This means we can ask the questions: What is the probability of obtaining a & c given column totals of a + b and c + d, and the grand total n?

$$Probability\ (a,c) = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{\frac{(a+b)!}{a!\,b!}\frac{(c+d)!}{c!\,d!}}{\frac{n!}{(a+c)!\,(b+d)!}} =$$

This is the probability of this configuration of the table and it is exact!

	A	Not A	*Column Total*
In Sample	a	b	*a+b*
Not In Sample	c	d	*c+d*
Row Total	*a+c*	*b+d*	*n*

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

The probability of this particular outcome is given by

$$p = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!}$$

107

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

Example: Suppose we want to examine whether intra-muscular magnesium is better than placebo for the treatment of chronic fatigue syndrome.

H_0: Magnesium Has No Effect
H_1: Magnesium Has An Effect

There are exactly 16 configurations of this table that could have been realized. Some more extreme than this one and some less extreme.

TABLE 1

	Magnesium	Placebo	Total
Felt better	12	3	15
Did not feel better	3	14	17
Total	15	17	32

108

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

Other Configurations

This test requires that the row and column totals stay the same in each configuration.

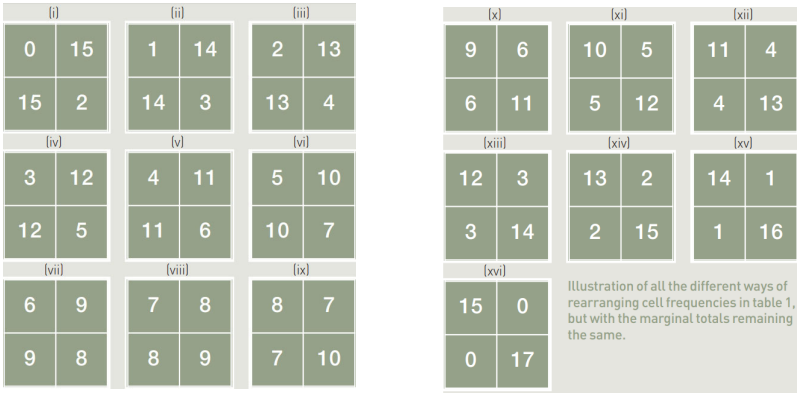


Illustration of all the different ways of rearranging cell frequencies in table 1, but with the marginal totals remaining the same.

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

Probabilities of Each Configurations

Total	*a*	*b*	*c*	*d*	*P*-value
i	0	15	15	2	0.0000002
ii	1	14	14	3	0.0000180
iii	2	13	13	4	0.0004417
iv	3	12	12	5	0.0049769
v	4	11	11	6	0.0298613
vi	5	10	10	7	0.1032349
vii	6	9	9	8	0.2150728
viii	7	8	8	9	0.2765221

Total	*a*	*b*	*c*	*d*	*P*-value
ix	8	7	7	10	0.2212177
x	9	6	6	11	0.1094916
xi	10	5	5	12	0.0328475
xii	11	4	4	13	0.0057426
xiii	12	3	3	14	0.0005469
xiv	13	2	2	15	0.0000252
xv	14	1	1	16	0.0000005
xvi	15	0	0	17	0.0000000

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

Probability of obtaining a result at least as extreme as XIII =

2 x (0.0005469 +
0.0000252 +
0.0000005 +
0.0000000)
= 0.001146

Total	*a*	*b*	*c*	*d*	*P*-value
ix	8	7	7	10	0.2212177
x	9	6	6	11	0.1094916
xi	10	5	5	12	0.0328475
xii	11	4	4	13	0.0057426
xiii	12	3	3	14	0.0005469
xiv	13	2	2	15	0.0000252
xv	14	1	1	16	0.0000005
xvi	15	0	0	17	0.0000000

Conclusion: Reject H_0 of No Effect

All Fisher tests are two-tailed test.

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

Example: Our data concern the distribution of eight ants' nests over 10 trees of each of two species of tree (A and B). There are two categorical explanatory variables (ants and trees), and four contingencies, ants (present or absent) and trees (A or B). The response variable is the vector of four counts `c(6,4,2,8)` entered columnwise:

	Tree A	Tree B	Row totals
With ants	6	2	8
Without ants	4	8	12
Column totals	10	10	20

We can calculate the probability for this particular outcome:

```
factorial(8)*factorial(12)*factorial(10)*factorial(10)/
      (factorial(6)*factorial(2)*factorial(4)*factorial(8)*factorial(20))
```

```
[1] 0.07501786
```

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

We need to compute the probability of outcomes that are *more extreme* than this.

There are two of them.

Suppose only 1 ant colony had been found on tree B. Then the table values would be 7, 3, 1, 9 but the row and column totals would be exactly the same (*the marginal totals are constrained*). The numerator always stays the same, so this case has probability

```
factorial(8)*factorial(12)*factorial(10)*factorial(10)/
      (factorial(7)*factorial(3)*factorial(1)*factorial(9)*factorial(20))

[1] 0.009526078
```


Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

There is an even more extreme case if no ant colonies at all were found on tree B. Now the table elements become 8, 0, 2, 10 with probability

```
factorial(8)*factorial(12)*factorial(10)*factorial(10)/
      (factorial(8)*factorial(2)*factorial(0)*factorial(10)*factorial(20))
[1] 0.0003572279
```

and we need to add these three probabilities together:

```
0.07501786 + 0.009526078 + 0.000352279
[1] 0.08489622
```

Contingency Tables with Small Expected Frequencies: Fisher's Exact Test

But there was no *a priori* reason for expecting that the result would be in this direction. It might have been tree A that happened to have relatively few ant colonies. We need to allow for extreme counts in the opposite direction by doubling this probability (all Fisher's exact tests are two-tailed):

```
2*(0.07501786 + 0.009526078 + 0.000352279)
[1] 0.1697924 #P-Value
```

This shows that there is no evidence of any correlation between tree and ant colonies. The observed pattern, or a more extreme one, could have arisen by chance alone with probability $p = 0.17$.

Fisher Test R Function

There is a built-in function called `fisher.test`, which saves us all this tedious computation. It can take as its argument a 2 × 2 (or larger matrix) containing the counts of the cells.

```
>x <- as.matrix(c(6,4,2,8))
>dim(x) <- c(2,2)
>fisher.test(x)

        Fisher's Exact Test for Count Data
data: x
p-value = 0.1698
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.6026805 79.8309210
sample estimates:
odds ratio
5.430473
```

Fisher Test R Code

```
>table <- read.table("c:\\temp\\fisher.txt",header=TRUE)
>head(table)
>attach(table)
>fisher.test(tree,nests)

        Fisher's Exact Test for Count Data
data: tree and nests
p-value = 0.1698
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
0.6026805 79.8309210
sample estimates:
odds ratio
5.430473
```

The `fisher.test` procedure can be used with matrices much bigger than 2 × 2.

Homework: Fisher Test

1. Which of the following are true with respect to Fisher's exact test for a contingency table:
   A. Applies to 2 x 2 contingency tables
   B. H_0: There is no association between rows and columns
   C. Appropriate when cell frequencies are small (< 5)
   D. The exact p-values can be calculated
   E. All of the above

2. Fisher's exact test is most appropriate when
   A. only the row marginal frequencies are fixed
   B. only the column marginal frequencies are fixed
   C. both marginal frequencies are fixed
   D. neither marginal frequencies is fixed

3. (T/F) Fisher Exact Test probabilities are based on the Poisson distribution.

Homework: Fisher Test

4. A researcher is comparing an experimental group (E) to a control group (C). The scores are shown below. The experimenter computes the median of all scores and considers everyone's performance that is above the median a success and everyone's below the median a failure. Compute Fisher's exact test to see if the difference in success rates is significant. Write an R program read in the data, calc the medians for each group, define the elements of the contingency table, and calculate the probability the table configuration. Also, perform the fisher test on the contingency table to validate you program results.

E<-c(10, 12, 13, 14, 16, 21)
C<-c(1, 3, 7, 8, 9, 11)

	E	C	*Column Total*
Success	a	b	*a+b*
Failure	c	d	*c+d*
Row Total	*a+c*	*b+d*	*n*

Hints:
1. You need to calc the median of each group
2. Those above the median are successes
3. Those below failures
4. The results will form a 2 x 2 contingency table
5. n = 12

Correlation and Covariance ★★★★★

Covariance is a measure used to determine how much two variables change in tandem or ***covary*** *or* ***move together***.

Correlation between two variables is a normalized version of the covariance. It is normalized by the standard deviation of each term so that its values is between -1 and 1.

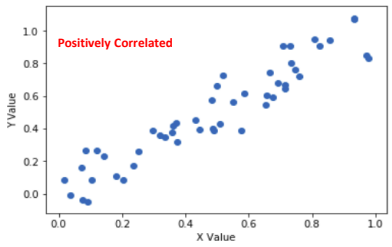
$$\mathrm{Corr}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_x \sigma_y}$$

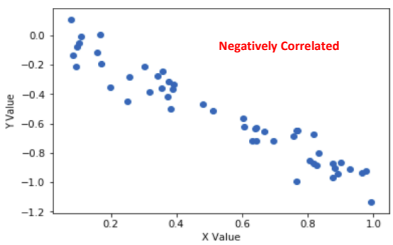
Covarianced normalized by Standard Deviation

Correlation between X and Y

Standard deviation of X

Standard deviation of Y



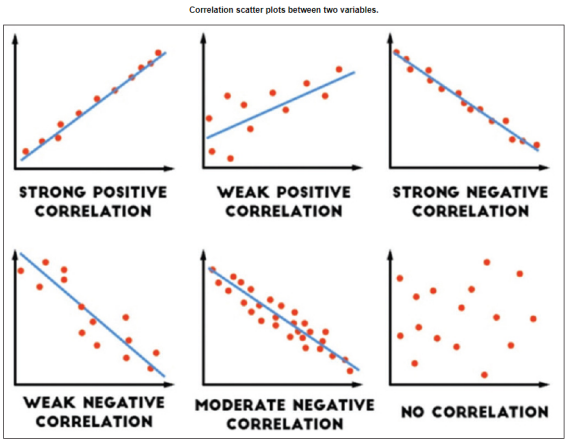


Correlation and Covariance ★★★★★

$$r = \frac{\operatorname{cov}(x,\, y)}{\sqrt{s_x^2 s_y^2}}$$

Alternate form of previous formula.

Correlation is a normalized form of covariance and not affected by scale. Both covariance and correlation measure the linear relationship between variables but cannot be used interchangeably.



Mathematical Expectation of Covariance ★★★★★

$$\begin{aligned}
\operatorname{cov}(X,Y) &= \mathrm{E}[(X - \mathrm{E}[X])\,(Y - \mathrm{E}[Y])] \\
&= \mathrm{E}[XY - X\,\mathrm{E}[Y] - \mathrm{E}[X]Y + \mathrm{E}[X]\,\mathrm{E}[Y]] && \text{<= Multiply (X – E[X])(Y – E[Y])} \\
&= \mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y] - \mathrm{E}[X]\,\mathrm{E}[Y] + \mathrm{E}[X]\,\mathrm{E}[Y] && \text{<= Apply Expectation Operator} \\
&= \mathrm{E}[XY] - \mathrm{E}[X]\,\mathrm{E}[Y] && \text{<= Apply Expectation Operator}
\end{aligned}$$

Note: **If X and Y are independent, then E[XY] = E[X]E[Y] and E[XY] – E[X]E[Y] = 0. This means cov(X, Y) = 0 and cor(X, Y) = 0.**

cor() is the R function for correlation.

Mathematical Expectation of Covariance

Let us work through a numerical example with data set twosample.txt. This is an interesting dataset. It has four columns:

1. x
2. y
3. a
4. b

No real meaning, but interesting values on the variables. We have gender on one of them.

```
> str(twosample)
'data.frame':   49 obs. of  4 variables:
 $ x: num  5.37 6.44 7.83 7.59 5.38 ...
 $ y: num  26.8 46.9 34.1 45.5 33.2 ...
 $ a: Factor w/ 5 levels "five","four",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ b: Factor w/ 2 levels "female","male": 2 2 1 1 2 1 1 2 2 2 ...
>
```

This is a 49 x 4 dimensional dataset.

	x	y	a	b
1	5.366516	26.76595	one	male
2	6.435778	46.89376	one	male
3	7.831232	34.11415	one	female
4	7.587142	45.49667	one	female
5	5.380939	33.22162	one	male
6	8.254098	39.98920	one	female
7	10.556489	24.43327	one	female
8	12.336669	49.61092	one	male
9	10.487217	39.29021	one	male
10	12.014185	39.63691	one	male
11	13.369883	44.54903	two	male
12	12.677217	46.51998	two	male
13	15.004940	43.16512	two	male
14	15.571449	55.06652	two	male
15	19.254838	58.70913	two	female
16	18.822350	72.85045	two	male
17	21.358761	74.71627	two	male
18	21.449881	78.92278	two	male
19	21.341156	71.08121	two	male

Mathematical Expectation of Covariance

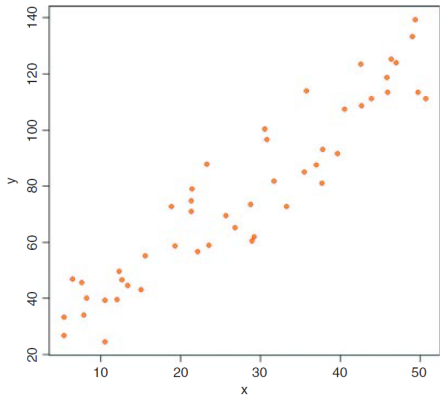
Let us work through a numerical example:

```
data <- read.table("c:\\temp\\twosample.txt",header=T)
attach(data)
plot(x, y, pch=21,col="red",bg="orange")
```

Plot Character = 21

Outline color = Red

Background or Interior Color = Orange



There is clearly a strong positive correlation between the two variables.

Mathematical Expectation of Covariance

Let complete the calculations:

```
var(x)
[1] 199.9837
```

0.15(0.06–0.082)(0.04–0.04975)+0.6(0.08–0.082)(0.05–0.04975)+0.25(0.10–0.082)(0.055–0.04975)

```
var(y)
[1] 977.0153
```

The covariance of *x* and *y*, cov(*x*, *y*), is given by the var function when we supply it with two vectors like this:

```
var(x,y)
[1] 414.9603
```


Mathematical Expectation of Covariance ★★★★★

Let complete the calculations:

Thus, the correlation coefficient should be 414.96/√(199.98 × 977.02)

```
var(x,y)/sqrt(var(x)*var(y))
[1] 0.9387684
```

Let us see if this checks out:

```
cor(x,y)
[1] 0.9387684
```

Note: The correlation coefficient was calculated two ways. One using the mathematical formula and the other using the R function.

Correlation Effect on Variance.

The more two variables covary, the less the variance is between them. This should make a lot of sense since variance measures differences. This happens a lot in paired data samples. The upstream/downstream invertebrate biodiversity data we looked at earlier showed this pattern.

This was the streams data where biodiversity (i.e., the variety of life in an ecosystem) was measured upstream and downstream in the same river. Therefore, we might expect some correlation between the pairs of data because it is the same river.

Correlation Effect on Variance

Example: The following data show the depth of the water table (in centimeters below the surface) in winter and summer at 10 locations

```
data <- read.table("c:\\temp\\wtable.txt",header=T)
attach(data)
names(data)
```
[1] "summer" "winter“

```
cor(summer, winter)
```
[1] 0.6596923

Correlation Effect on Variance

Example: The following data show the depth of the water table (in centimeters below the surface) in winter and summer at 10 locations

```
data <- read.table("c:\\temp\\wtable.txt",header=T)
attach(data)
names(data)
```
[1] "summer" "winter"

```
cor(summer, winter)
```
[1] 0.9720949

There is a strong positive correlation Not surprisingly, places where the water table is high in summer tend to have a high-water table in winter as well.

Statistical Significance of Correlation Coefficient ★★★★★

If you want to determine the significance of a correlation (i.e. the *p* value associated with the calculated value of *r*) then use `cor.test` rather than `cor`. This test has non-parametric options for Kendall's *tau* or Spearman's rank, depending on the method you specify (`method="k"` or `method="s"`), but the default method is Pearson's product-moment correlation (`method="p"`):

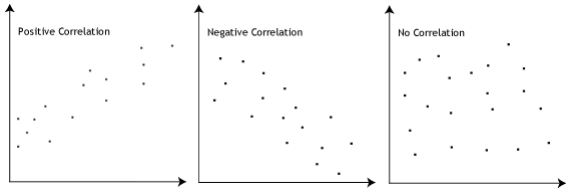
```
cor.test(summer, winter)

        Pearson's product-moment correlation
data:  summer and winter
t = 11.721, df = 8, p-value = 2.565e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8827796 0.9935887
sample estimates:
      cor
0.9720949
```

Pearson's Product-Moment Correlation ★★★★★

It is given by the formula:

$$\text{Corr}(X,Y) = \frac{\text{Cov}(X,Y)}{\sigma_x \sigma_y}$$

- Corr(X,Y): Correlation between X and Y
- Cov(X,Y): Covarianced normalized by Standard Deviation
- σ_x: Standard deviation of X
- σ_y: Standard deviation of Y



Homework: Covariance & Correlation

1. Use r to calculate the correlation coefficient for the bivariate data (x_i, y_i) : (0, 0)(3, 1.4)(6 ,2.6)(7, 3.8)(9, 7.2)
2. A correlation coefficient of r = 0.8 is reported for a sample of pairs (x_i, y_i) . Without any further information, this implies that: Exactly one option must be correct)
   A. As the x values decrease, the y values increase.
   B. 80% of the variation in y is due to regression on x.
   C. The (x_i, y_i) are scattered about a straight line of unknown positive slope.
   D. The (x_i, y_i) are scattered about a straight line of slope 0.8.
3. The correlation coefficient for a set of bivariate data (x_i, y_i) is r = 0.87, where the xi are measured in inches and the yi are measured in lbs. A second analyst records the xi values in cm. (1 inch ≈ 2.5 cm). What is the second analyst's value of the correlation coefficient? Exactly one option must be correct)
   A. 0.35
   B. 0.87
   C. 2.18
   D. Unable to determine without knowing the yi units.

Homework: Covariance & Correlation

4. Use the following data to answer this question:

   - We anticipate that there is a 15% chance that next year's stock returns for ABC Corp will be 6%, a 60% probability that they will be 8% and a 25% probability that they will be 10%. We already know the expected value of returns is 8.2% and the standard deviation is 1.249%.
   - We also anticipate that the same probabilities and states are associated with a 4% return for XYZ Corp, a 5% return, and a 5.5% return. The expected value of returns is then 4.975 and the standard deviation is 0.46%.

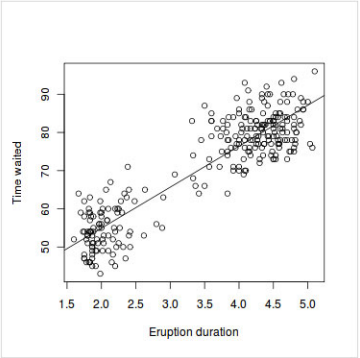
   Calculate the covariance and correlation between the returns of ABC and XYZ using R. What is your conclusion about the returns of the two companies?

Homework: Covariance & Correlation

5. Use the built-in data frame faithful to answer the questions below. There are two observation variables in the data set. The first one, called eruptions, is the duration of the geyser eruptions. The second one, called waiting, is the length of waiting period until the next eruption.

   A. Create a scatter plot of the relationship between these two variables.
   B. Add a line of best fit to the scatter plot.
   C. Calculate the covariance and correlation between eruption and waiting.
   D. How would you explain the two clusters of points?

Your code should replicate this graph ->



Homework: Covariance & Correlation

6. Consider a sample of 60 observations on variables X and Y in which the correlation is 0.42. If the critical t-values are +1.994, we

   A. conclude that there is little correlation between X and Y.
   B. conclude that there is no significant correlation between X and Y.
   C. cannot test the significance of the correlation with this information.
   D. conclude that there is statistically significant correlation between X and Y

7. (T/F) Correlations maybe scale dependent if they are oppositely correlated on a small and large scale.

135

Kolmogorov–Smirnov Test

This is an extremely simple test for asking one of two different questions:

1. Are two sample distributions the same, or are they significantly different from one another in one or more (unspecified) ways?

2. Does a particular sample distribution arise from a particular hypothesized distribution?

136

Kolmogorov–Smirnov (K-S) Test

Two samples can differ in the following ways:

1. Means
2. Variances
3. Skewness
4. Kurtosis

Even if they have exact means and significantly differ on the other measures, then they are assumed to come from different distributions.

The Kolmogorov–Smirnov test works on cumulative distribution, which reflects all four metrics above.

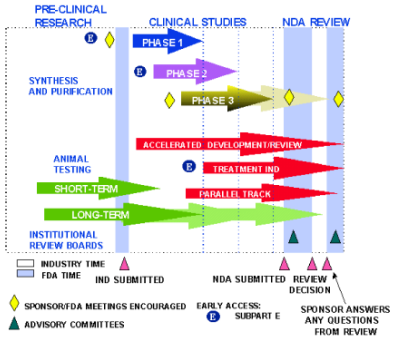
Kolmogorov–Smirnov (K-S) Test ★★★★★

Cumulative distribution functions give the probability that a randomly selected value of X is less than or equal to x:

$$F(x) = P[X \leq x]$$

The K-S Test compares cumulative distribution functions for two different samples to test how difference they are. If they are different then the two samples are assumed to come from two different distributions

This is important in comparing the effect of a group getting a drug, for example, and a group that gets the placebo. Experiments of this type are the basis of FDA drug testing. This results of a K-S test would be used to determine if the drug is an effective treatment for a condition.



Kolmogorov–Smirnov (K-S) Test ★★★★★

Example: Suppose we had insect wing sizes (*y*) for two geographically separated populations (A and B) and we wanted to test whether the distribution of wing lengths was the same in the two places:

```
data <- read.table("c:\\temp\\ksdata.txt",header=T)
attach(data)
names(data)
[1] "y" "site"
```

We start by extracting the data for the two populations, and describing the samples:

```
table(site)
site
A B
10 12
```


Kolmogorov–Smirnov (K-S) Test

```
tapply(y,site,mean)
A B
4.355266 11.665089

tapply(y,site,var)
A B
27.32573 90.30233
```

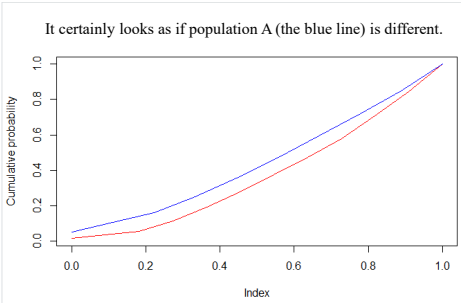
Observations:

- Their means are quite different.
- The size of the difference in their variances is to large for the t-test, which requires equal variances.

Kolmogorov–Smirnov (K-S) Test

Let's plot their cumulative distribution functions:

```
data <- ksdata
attach(data)
names(data)

plot(seq(0,1,length=12),cumsum(sort(B)/sum(B)),type="l",
   ylab="Cumulative probability",xlab="Index",col="red")

lines(seq(0,1,length=10),cumsum(na.omit(sort(A))/sum(na.omit(A))),col="blue")
```



This dataset presented some unique problems because A only had 10 observations while B has 12. When imported, R put "NA" in two fields, and they prevent summations unless we tell R to omit them from the calculations. We can do this with the "na.omit" function.

★★★★★

Kolmogorov–Smirnov (K-S) Test

We test the significance of the difference between the two distributions with ks.test like this:

```
ks.test(A, B)

        Two-sample Kolmogorov-Smirnov test

data:  A and B
D = 0.35, p-value = 0.5161
alternative hypothesis: two-sided

Warning message:
In ks.test(A, B) : cannot compute exact p-
value with ties
```

Kolmogorov–Smirnov (K-S) Test – Checking Against a Normal Distribution

We test the significance of the difference between the two distributions with ks.test like this:

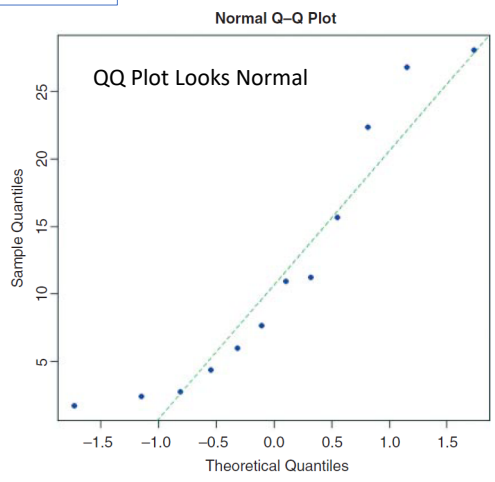
```
ks.test(B,"pnorm", A)

        One-sample Kolmogorov-Smirnov test

data:  B
D = NA, p-value = NA
alternative hypothesis: two-sided
```

There is no evidence that the samples from site B depart significantly from normality.

The Shapiro Test

```
shapiro.test(B)

        Shapiro-Wilk normality test

data:  B
W = 0.93936, p-value = 0.4898
```

```
qqnorm(B, pch=16,col="blue")
qqline(B, col="green",lty=2)
```



Bootstrapping – What is it?

The bootstrap method is a resampling technique used to estimate statistics on a population by sampling a dataset with replacement.

It can be used to estimate summary statistics such as the mean or standard deviation. It is used in applied machine learning to estimate the skill of machine learning models when making predictions on data not included in the training data.

Bootstrapping – What is the Process?

The bootstrap method can be used to estimate a quantity of a population. This is done by repeatedly taking small samples, calculating the statistic, and taking the average of the calculated statistics. We can summarize this procedure as follows:

1. Choose a number of bootstrap samples to perform
2. Choose a sample size
3. For each bootstrap sample
    1. Draw a sample with replacement with the chosen size
    2. Calculate the statistic on the sample
4. Calculate the mean of the calculated sample statistics.

Bootstrapping

Example: We want to use bootstrapping to obtain a 95% confidence interval for the mean of a vector of numbers called values:

```
data <- read.table("c:\\temp\\skewdata.txt",header=T)
attach(data)
names(data)

[1] "values"
```


Bootstrapping

We shall sample with replacement from values using `sample(values,replace=T)`, then work out the mean, repeating this operation 10 000 times, and storing the 10 000 different mean values in a vector called `ms`:

```
ms <- numeric(10000)

for (i in 1:10000){
                    ms[i] <- mean(sample(values, replace=T))
                  }
```

Loop

Bootstrapping

We shall sample with replacement from values using `sample(values,replace=T)`, then work out the mean, repeating this operation 10 000 times, and storing the 10 000 different mean values in a vector called `ms`:

```
ms <- numeric(10000)

for (i in 1:10000){
                    ms[i] <- mean(sample(values, replace=T))
                  }
```

Loop

Notice: Everytime the function is called it pulls a sample size equal to the entire data set in this case 30. We can change this with the size argument –sample(x, **size**, replace = FALSE, prob = NULL).

Bootstrapping

To get the 95% confidence interval, we need the quantiles for 0.025 and 0.975

```
quantile(ms,c(0.025,0.975))
2.5%          97.5%
24.97918      37.62932
```

```
mean(values)
[1] 30.96866
```

Thus the intervals below and above the mean are

```
mean(values)-quantile(ms,c(0.025,0.975))
2.5%          97.5%
5.989472   -  6.660659
```

However, the results differs from the parametric values because of skewness!

The boot Package

```
install.packages("boot")
library(boot)
```

The syntax of `boot` is very simple:

```
boot(data, statistic, R)
```

The trick to using `boot` lies in understanding how to write the statistic function.

151

The boot Package

The syntax of `boot` is very simple:

```
boot(data, statistic, R)
```

`R` is the number of resamplings you want to do (`R=10000` in this example)
`data` is the name of the data object to be resampled (`values` in this case)

The second argument is an index (a vector of subscripts) that is used within `boot` to select random assortments of `values`.

Our `statistic` function can use the built-in function `mean` to calculate the mean value of the sample of `values`.

```
mymean <- function(values,i) mean(values[i])
```

← Syntax for built-in functions

152

The boot Package

The syntax of `boot` is very simple:

```
boot(data, statistic, R)
```

Now we can run the bootstrap for 10,000 iterations:

```
myboot <- boot(values, mymean, R=10000)
myboot

ORDINARY NONPARAMETRIC BOOTSTRAP
Call:
boot(data = values, statistic = mymean, R = 10000)
Bootstrap Statistics :
original Bias std. error
t1* 30.96866 -0.08155796 3.266455
```


153

The boot Package

`bias` is the difference between the arithmetic mean and the mean of the bootstrapped samples which are in the variable called `myboot$t:`

```
mean(myboot$t)-mean(values)
[1] -0.08155796
```

and **`std. error`** is the standard deviation of the simulated values in `myboot$t`:

```
sqrt(var(myboot$t))
[,1]
[1,] 3.266455
```


154

The boot Package

The output is interpreted as follows. The `original` is the mean of the whole sample:

```
mean(values)
[1] 30.96866
```

`bias` is the difference between the arithmetic mean and the mean of the bootstrapped samples which are in the variable called `myboot$t`:

```
mean(myboot$t)-mean(values)
[1] -0.08155796
```

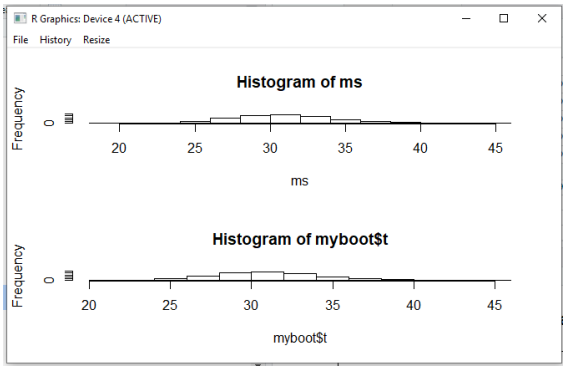
`std. error` is the standard deviation of the simulated values in `myboot$t`:

```
sqrt(var(myboot$t))
[,1]
[1,] 3.266455
```


The boot Package

The components of `myboot` can be used to do other things. For instance, we can compare our homemade vector (`ms` above) with a histogram of `myboot$t`:

```
windows(7,4)
par(mfrow=c(2,1))
hist(ms)
hist(myboot$t)
```



The boot Package

They differ in detail because they were generated with different series of random numbers. Here are the 95% intervals for comparison with ours, calculated from the quantiles of `myboot$t`:

```
mean(values)-quantile(myboot$t,c(0.025,0.975))
2.5%          97.5%
6.126120   -   6.599232
```


157

The boot Package

There is a function `boot.ci` for calculating confidence intervals from the `boot` object:

```
boot.ci(myboot)
```

```
BOOTSTRAP CONFIDENCE INTERVAL CALCULATIONS
Based on 10000 bootstrap replicates
CALL :
boot.ci(boot.out = myboot)
Intervals :
    Level      Normal         Basic
    95%        (24.65, 37.45)         (24.37, 37.10)
    Level      Percentile             BCa
    95%        (24.84, 37.57)         (25.63, 38.91)
Calculations and Intervals on Original Scale
Warning message:
bootstrap variances needed for studentized intervals in:
boot.ci(myboot)
```


158

The boot Package

`Normal` is the parametric CI based on the standard error of the mean and the sample size.

The `Percentile` interval is the quantile from the bootstrapped estimates:

```
quantile(myboot$t, c(0.025,0.975))
      2.5%          97.5%
      24.84254      37.56789
```

which, as we saw earlier, was close to our home-made values (above).

The `BCa` interval is the bias-corrected accelerated percentile. It is the interval preferred by statisticians

HW: Bootstrapping

Write a program where the first lines at code are as follows:

```
install.packages("boot",dep=TRUE)
library(boot)
hsb2 <- read.table("https://stats.idre.ucla.edu/stat/data/hsb2.csv", sep=",", header=T)
 pearson <- function(d, i){
      d2 <- d[i, ]
return(cor(d2$write, d2$math))
```

This is the built-in function for the Pearson correlation coefficient.

Next write additional code to perform the following:

1. Resample the data 500 times using the bootcorr command
2. Print the summary
3. Calculate the bias (-0.001528707) and standard error (0.04020362) using bootcorr$t and bootcorr$t0
4. Create a plot using >plot(bootcorr)