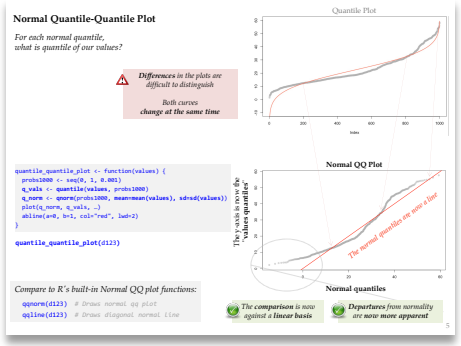
Business Analytics Using
Computational Statistics

Week 6
Permutation Tests

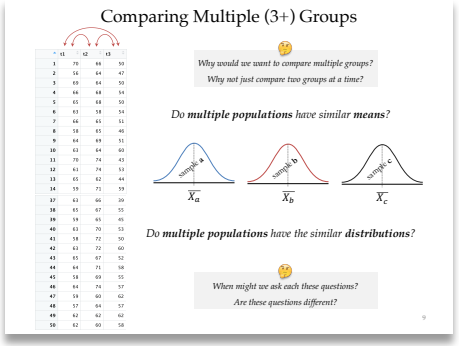
Week 7
Multigroup Comparisons

Week 8
No Class!

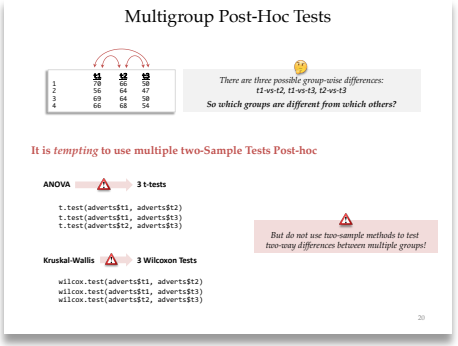
Visualizing Normality



Multigroup Tests



Posthoc Tests



Picking Packages

TidyR

Well maintained and updated

Simpler Syntax

Used in many books/tutorials

Part of larger **"tidy" universe**

(might be better for large analytics project)

Reshape2

Seems to be "superseded" by TidyR

(will it stop being maintained one day?)

Works on more than data.frames

Still found in online tutorials

Custom built for reshaping

(might be better for using in your own packages)

*There's a **lot of decisions** that should go into adopting a package for your work.*

*Consider the factors that are **important for you***

There are no right or wrong choices

(you may learn more by switching between packages)

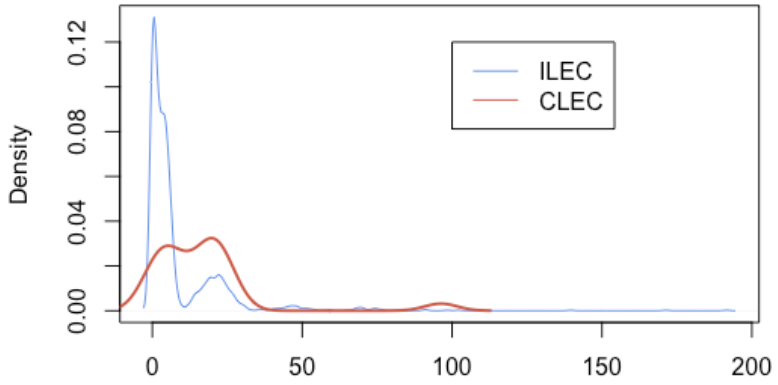
```
library(tidyR)
verizon <- read.csv("verizon_wide.csv")

verizon_long <- gather(verizon_wide, na.rm = TRUE, key = "provider", value = "time")
providers <- split(x = verizon_long$time, f = verizon_long$provider)
```

t-Test of Sample Means

Assumptions of Two-Sample t-Tests:

1. Ratio/Interval scale (continuous numbers)
2. Normal distribution of data
3. Homogeneity of variance (constant variance)



Student's t-Test assuming equal variances

```
t.test(providers$CLEC, providers$ILEC, alt="greater", var.equal=TRUE, conf.level = 0.99)

# t = 2.6125, df = 1685, p-value = 0.004534
```

Reject H_{null}

Assumptions about variances are important:
we arrive at contrary conclusions at $\alpha = 0.01$

Welch's t-Test assuming unequal variances

```
t.test(providers$CLEC, providers$ILEC, alt="greater", var.equal=FALSE, conf.level = 0.99)

# t = 1.9834, df = 22.346, p-value = 0.02987
```

Cannot Reject H_{null}

We must also confirm normality assumption

Assessing Normality

How can we visually compare distributions?

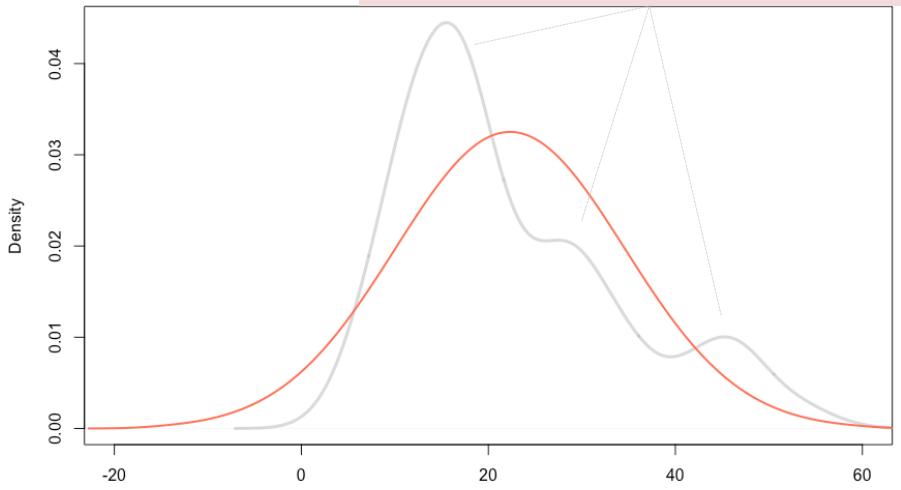
Density Plots

We can plot the distributions of our data against perfect normal

Let's take 1000 quantiles from a normal distribution with same ***central tendencey*** *and* ***dispersion*** *as our original data*

```
probs1000 <- seq(0, 1, 0.001)
qnorm1000 <- qnorm(probs1000, mean=mean(d123), sd=sd(d123))

plot(density(d123), main=NA, xlab=NA)
lines(density(qnorm1000))
```

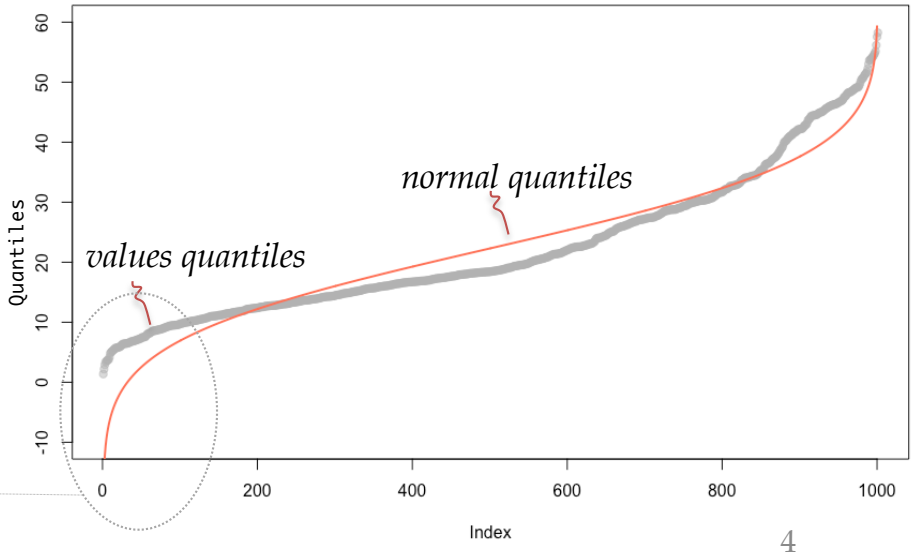


Our data has many distracting features that make comparisons hard

Quantile Plots

We can plot the quantiles of our data against perfect normal

```
plot(quantile(d123, probs=probs1000))
lines(qnorm1000)
```



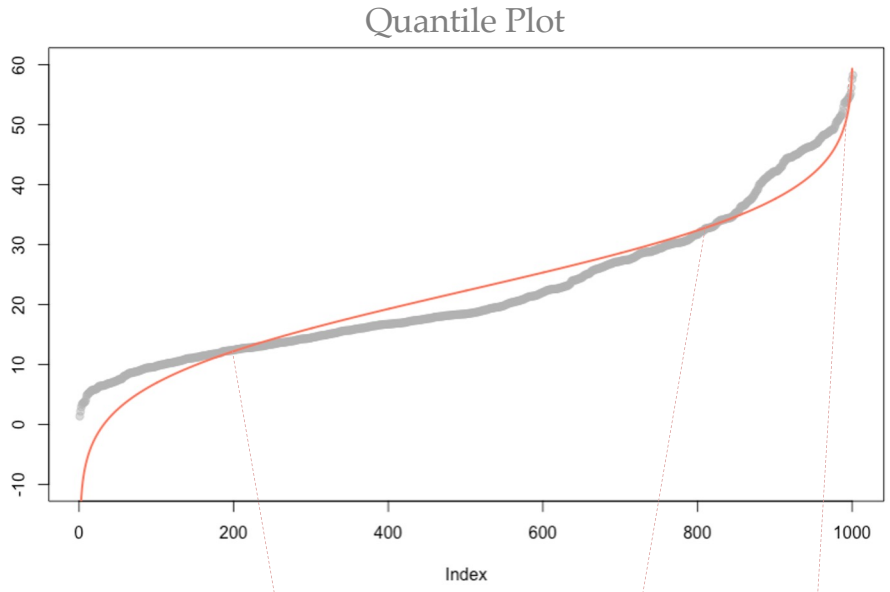
Removes many of the features that make comparisons difficult

Normal Quantile-Quantile Plot

For each normal quantile,
what is quantile of our values?

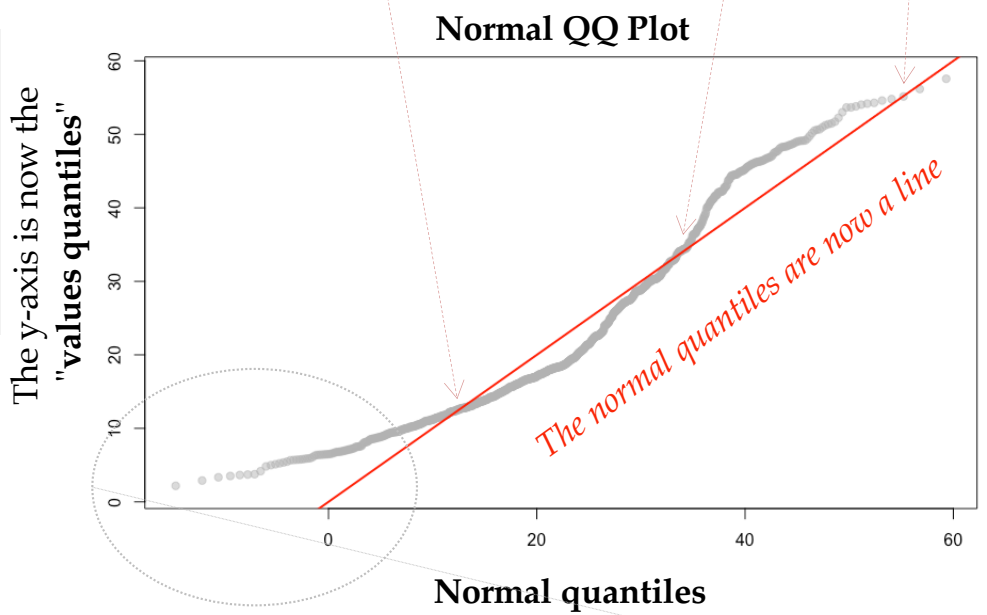
***Differences** in the plots are difficult to distinguish*

Both curves
change at the same time




```
quantile_quantile_plot <- function(values) {
  probs1000 <- seq(0, 1, 0.001)
  q_vals <- quantile(values, probs1000)
  q_norm <- qnorm(probs1000, mean=mean(values), sd=sd(values))
  plot(q_norm, q_vals, …)
  abline(a=0, b=1, col="red", lwd=2)
}
```

```
quantile_quantile_plot(d123)
```



Compare to R's built-in Normal QQ plot functions:

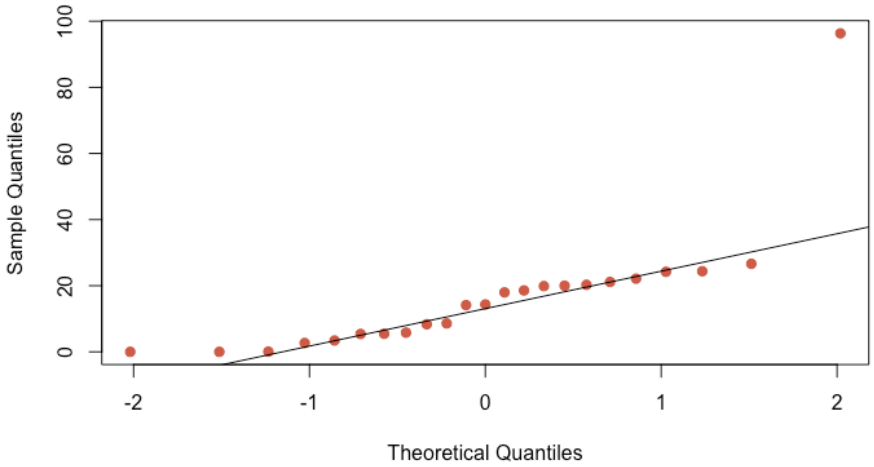
```
qqnorm(d123)   # Draws normal qq plot
qqline(d123)   # Draws diagonal normal line
```

*The **comparison** is now against a **linear basis***

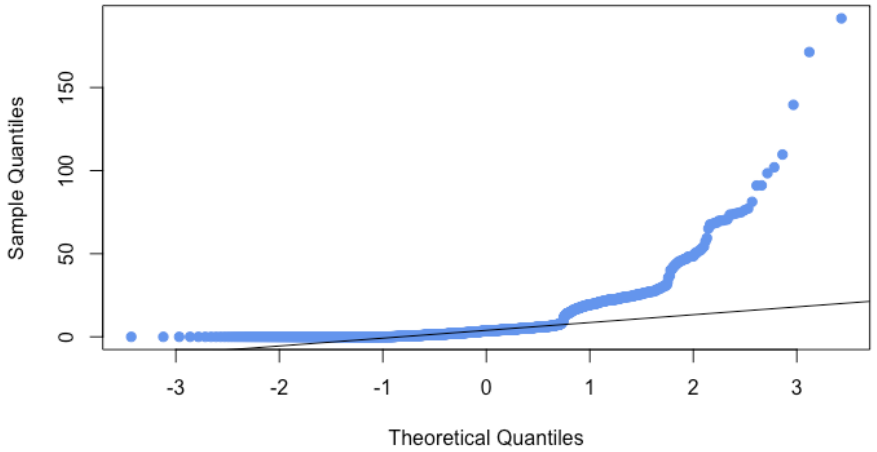
Departures** from normality are **now more apparent

Does our data match assumptions of normality?

```
qqnorm(providers$CLEC, col="coral3", pch=19)
qqline(providers$CLEC)
```




```
qqnorm(providers$ILEC, col="cornflowerblue", pch=19)
qqline(providers$ILEC)
```



*We should likely prefer our **non-parametric tests***

You now know exactly how Normal-QQ plots work,
But use the built-in functions `qqnorm()`, `qqline()` in your work

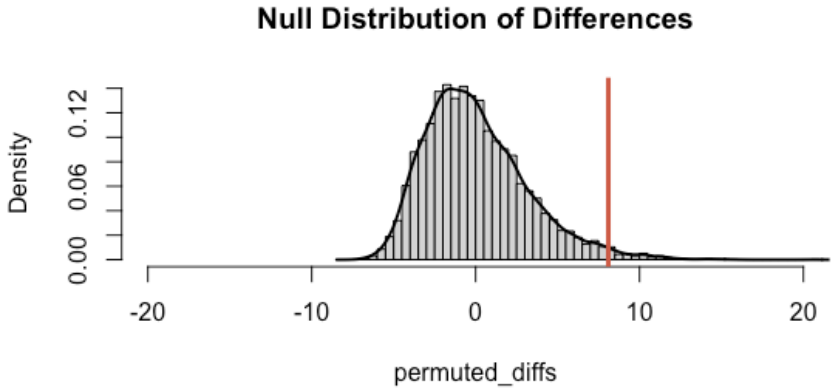
Permutation Tests of Sample Means

```r
observed_diff <- mean(providers$CLEC) - mean(providers$ILEC)

# Simulate Permuted differences
permute_diff <- function(values, groups) {
  permuted <- sample(values, replace = FALSE)
  grouped <- split(permuted, groups)
  mean(grouped$CLEC) - mean(grouped$ILEC)
}

nperms <- 10000
set.seed(895702709)
permuted_diffs <- replicate(nperms, permute_diff(verizon_long$time, verizon_long$provider))
```

```r
# Visualize permuted differences
hist(permuted_diffs, breaks = "fd", probability = TRUE, ...)
lines(density(permuted_diffs), lwd=2)
abline(v=observed_diff, col="coral3", lwd=3)
```




```r
# Significance of permuted differences
p_1tailed <- sum(permuted_diffs > observed_diff) / nperm
# [1] 0.0186
```

Cannot Reject H_{null}

Assumption free conclusion: ***we cannot reject*** H_{null}

Would two-tailed p-value be different?

Wilcoxon Test of Samples

```
gt_eq <- function(a, b) {
  ifelse(a > b, 1, 0) + ifelse(a == b, 0.5, 0)
}

W <- sum(outer(providers$CLEC, providers$ILEC, FUN = gt_eq))
# [1] 26820

n1 = length(providers$CLEC) # 23
n2 = length(providers$ILEC) # 1664

wilcox_p_1tail <- 1 - pwilcox(W, n1, n2)  # 0.0003688341
wilcox_p_2tail <- 2 * wilcox_p_1tail      # 0.0007376683

wilcox.test(providers$CLEC, providers$ILEC, alternative = "greater", paired = FALSE)
W = 26820, p-value = 0.0004565
```

Reject H_{null}

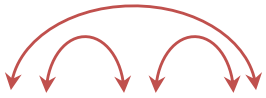
Assumption free conclusion:
Values in the two samples are distinct
(values seem shifted)

Mean of samples *are not distinguishable*

BUT

CLEC-vs-ILEC customers seem to be ***treated differently***

Comparing Multiple (3+) Groups



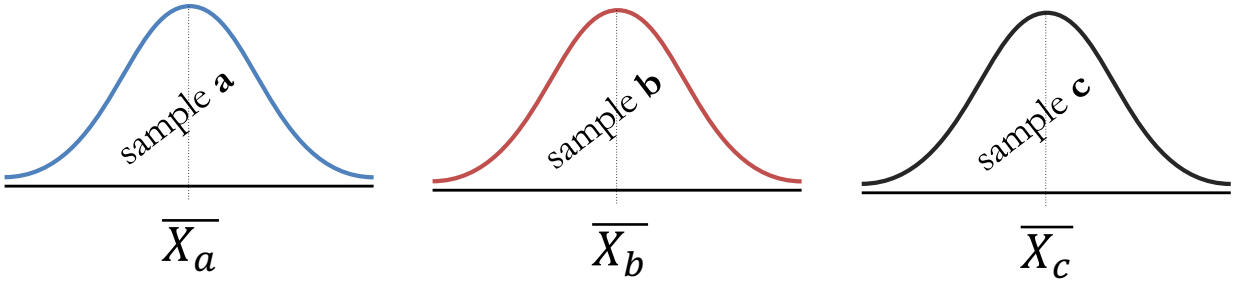
	t1	t2	t3
1	70	66	50
2	56	64	47
3	69	64	50
4	66	68	54
5	65	68	50
6	63	58	54
7	66	65	51
8	58	65	46
9	64	69	51
10	63	64	60
11	70	74	43
12	61	74	53
13	65	62	44
14	59	71	59
37	63	66	39
38	65	67	55
39	59	65	45
40	63	70	53
41	58	72	50
42	63	72	60
43	65	67	52
44	64	71	58
45	58	69	55
46	64	74	57
47	59	60	62
48	57	64	57
49	62	62	62
50	62	60	58

🤔

Why would we want to compare multiple groups?

Why not just compare two groups at a time?

*Do **multiple populations** have similar **means**?*



*Do **multiple populations** have the similar **distributions**?*

🤔

When might we ask each these questions?

Are these questions different?

Advertising Example

A marketing manager is trying to develop a new nation-wide advertising campaign. She is not sure whether to advertise the low price, the high quality, or the great features of the product. To find out which strategy works best, she conducts an experiment using three different advertisements in three different cities that have nearly identical sales and demographics. She monitors the sales in each city for 50 days.
Assume: the 3 strategies might not have run at the same time; the 3 cities are very similar; people did not share their reviews between cities

She treats the three advertisements differently:
- the *low product price* is advertised in the first city (t1)
- the *high product quality* is advertised in the second city (t2)
- the *number of product features is advertized* in the third city (t3)

1. ***Did the three strategies produce different sales, on average?***
2. ***Did the three strategies produce difference sales numbers?***

```
adverts <- read.table("marketing_strategies.txt", header=TRUE)
```

Descriptive Statistics:

```
sapply(adverts, mean)
   t1    t2    t3
62.36 66.74 51.96
```

```
sapply(adverts, sd)
      t1       t2       t3
4.279805 4.173825 6.630049
```

$\bar{x}_1 : 62.36$ $s_1 : 4.28$

$\bar{x}_2 : 66.74$ $s_2 : 4.17$

$\bar{x}_3 : 51.96$ $s_3 : 6.63$

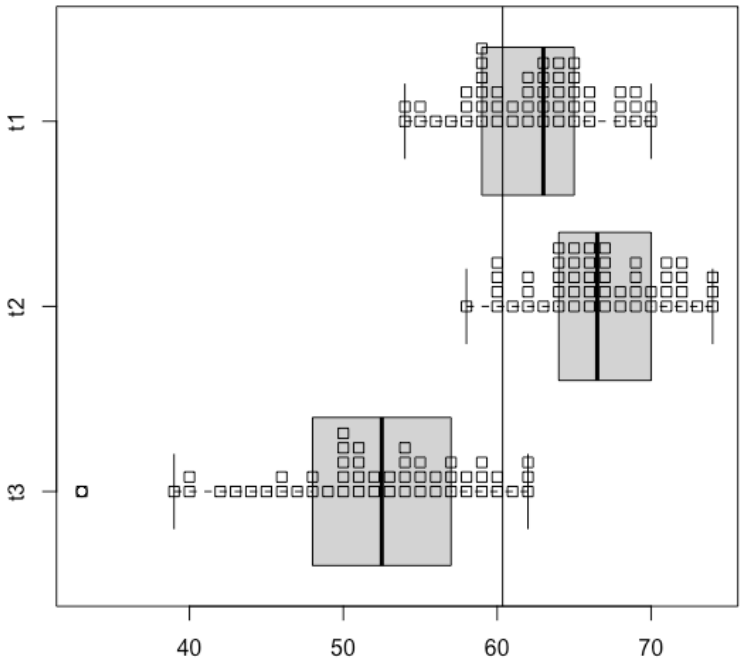
Overall sample mean (grand mean):

```
mean(sapply(adverts, mean))
[1] 60.35333
```

$\bar{\bar{x}} : 60.35$

`rev()` reverses order

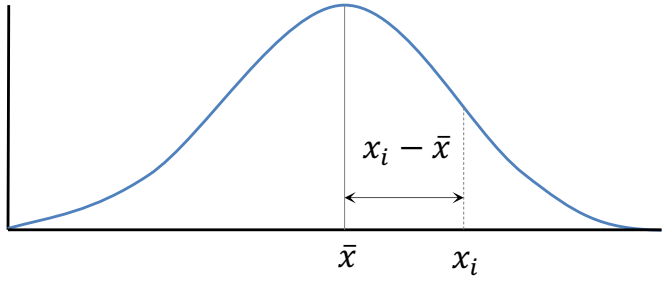
```
boxplot(rev(adverts), horizontal=TRUE)
stripchart(rev(adverts),method="stack", add=TRUE)
abline(v=mean(sapply(adverts, mean)))
```



wide dataframe

	t1	t2	t3
1	70	66	50
2	56	64	47
3	69	64	50
4	66	68	54
5	65	68	50
6	63	58	54
7	66	65	51
8	58	65	46
9	64	69	51
∘	∘	∘	∘
49	62	62	62
50	62	60	58

Parametric Approach: Analyze Variances



Absolute Deviation

$|x_i - \bar{x}|$

Mean Absolute Deviation

$$\frac{\sum|x_i - \bar{x}|}{N}$$

Variability

$$\sum(x_i - \bar{x})^2$$

"Sum of Squares"

Variance

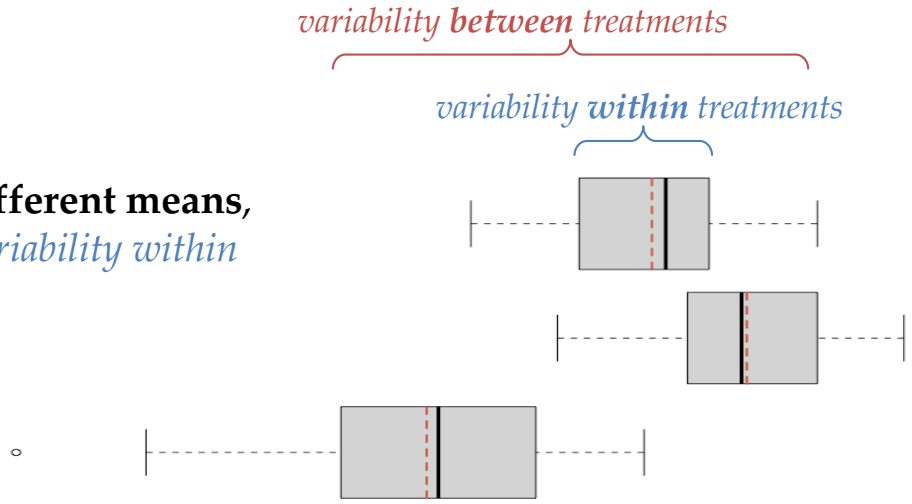
$$\frac{\sum(x_i - \bar{x})^2}{n-1}$$

"Mean Squares"
(average variability?)

Sources of Variance: Between / Within Treatments

Treatment: what was changed to create different groups *(e.g., advertising strategy)*

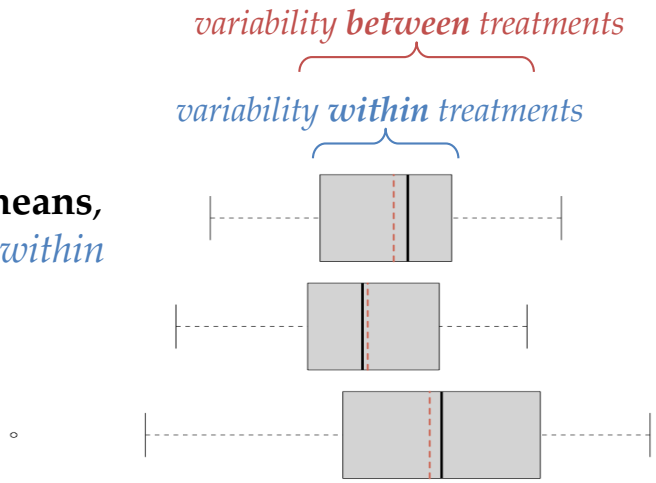
If "treatments" have **different means**,
variability between > *variability within*



Treatment variability
(variability between treatments)
+
Error variability
(variability within treatments)

TOTAL Variability

If "treatments" have **similar means**,
variability between ≈ *variability within*



*We can **analyze variabilities** to see if treatments were effective!*

One-way Analysis of Variance (ANOVA)

H_{null} : The means of the three treatment populations are the same $\mu_1 = \mu_2 = \mu_3$

H_{alt} : The means of the three treatments populations are not the same ***at least one mean is different***

k treatments of 1 factor

	t1	t2	t3
1	70	66	50
2	56	64	47
3	69	64	50
4	66	68	54

n_j *observations of each treatment*

$n_j \times k = n_T$ *total observaitons*

SSTR: sum of squares due to treatments **(treatment variability)**
MSTR: mean square due to treatments **(treatment variance)**

SSE: sum of squares due to error **(error variability)**
MSE: mean square due to error **(error variance)**

between-treatments variance:
(variance of treatment means from grand mean)

$$MSTR = \frac{SSTR}{df_{MSTR}} = \frac{\sum_{j=1}^{k} n_j(\bar{x}_j - \bar{\bar{x}})^2}{k-1}$$

within-treatments variance:
(average variance within treatments)

$$MSE = \frac{SSE}{df_{MSE}} = \frac{\sum_{j=1}^{k} (n_j - 1)s_j^2}{n_T - k}$$

$$F = \frac{MSTR}{MSE}$$

*Ratio of variances follows **F** distrbution*

k: number of treatments
n_j: sample size of treatment j
s_j: standard deviation of treatment j
n_T: size of all treatments combined

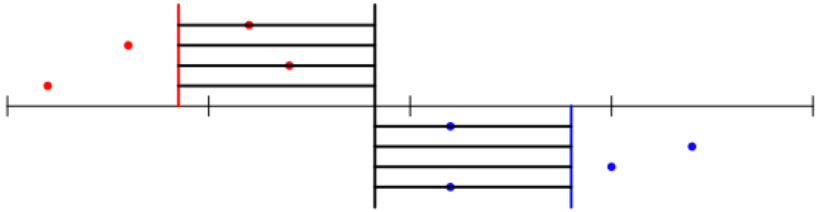
$\bar{x}_j$: mean of sample j
$\bar{\bar{x}}$: grand mean (mean of all sample means)
df : degrees of freedom

***between-treatments* estimate of variance**
"variance between treatment means"

s^2 = (treatment variance) + (error variance)

Will be larger than MSE if H_{null} is false

$$MSTR = \frac{SSTR}{df_{MSTR}} = \frac{\sum_{j=1}^{k} n_j(\bar{x}_j - \bar{\bar{x}})^2}{k-1}$$

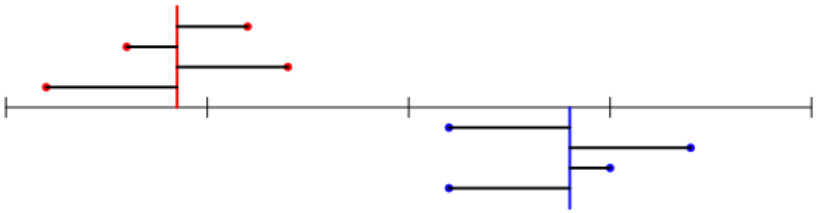


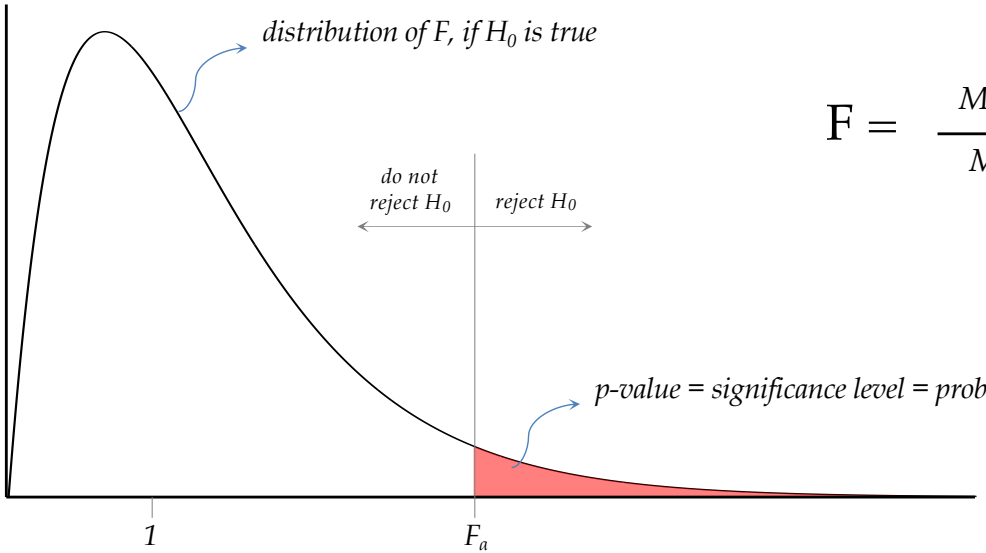
***within-treatments* estimate of variance**
"average variance of all treatments"

s^2 = (error variance)

$$MSE = \frac{SSE}{df_{MSE}} = \frac{\sum_{j=1}^{k} s_j^2}{k}$$ if treatment sizes are the same

$$MSE = \frac{SSE}{df_{MSE}} = \frac{\sum_{j=1}^{k} (n_j - 1)s_j^2}{n_T - k}$$ if treatment sizes are different





$$F = \frac{MSTR}{MSE} = \frac{\text{(treatment variance)} + \text{(error variance)}}{\text{(error variance)}}$$

One-way ANOVA the long way in R

functional iteration -- we don't need to explicitly loop through all the treatments!

$$MSTR = \frac{SSTR}{df_{MSTR}} = \frac{\sum_{j=1}^{k} n_j(\bar{x}_j - \bar{\bar{x}})^2}{k-1} = \frac{5763.21}{3-1} = 2881.61$$

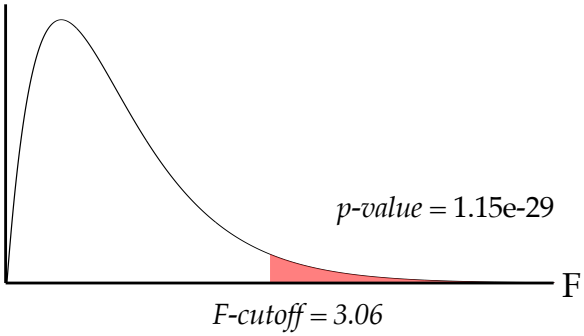
```
sstr <- 50*sum((sapply(adverts, mean) - mean(sapply(adverts, mean)))^2)
df_mstr <- 3-1
mstr <- sstr/df_mstr
```

$$MSE = \frac{SSE}{df_{MSE}} = \frac{\sum_{j=1}^{k}(n_j - 1)s_j^2}{n_T - k} = \frac{3905.06}{150-3} = 26.56$$

```
sse <- sum((50-1)*sapply(adverts, var))
df_mse <- 50*3 - 3
mse <- sse/df_mse
```

$$F = \frac{MSTR}{MSE} = \frac{2881.61}{26.56} = 108.47$$

```
f_value <- mstr/mse
```




```
qf(p=0.95, df1=df_mstr, df2=df_mse)
```

```
p_value <- pf(f_value, df_mstr, df_mse, lower.tail=FALSE)
```

One-way ANOVA the easy way in R

	t1	t2	t3
1	70	66	50
2	56	64	47
3	69	64	50
4	66	68	54

	strategy	sales
1	t1	70
2	t1	56
3	t1	69
4	t1	66
...	...	...
51	t2	66
52	t2	64
53	t2	64
54	t2	68
...	...	...
101	t3	50
102	t3	47
103	t3	50
104	t3	54
...	...	...

Change shape data of data to row-wise format

```
library(reshape2)
ads <- melt(adverts, id.vars = NULL,
            variable.name = "strategy",
            value.name = "sales")
```

Run `oneway.test()` function for one-way ANOVA

*The **tilde** ~ separates dependent and independent variables*

***`factor()`** tells R that this is a **categorical** variable*

```
oneway.test(ads$sales ~ factor(ads$strategy), var.equal=TRUE)

 One-way analysis of means

data:  ads$sales and ads$strategy
F = 108.4737, num df = 2, denom df = 147, p-value < 2.2e-16
```

Alternatively, run the more general `aov()` function for ANOVA

```
anova_model <- aov( ads$sales ~ factor(ads$strategy))
summary(anova_model)
```

***`summary()`** often gives a more in-depth report on an estimated model*

```
                      Df Sum Sq Mean Sq F value Pr(>F)
factor(ads$strategy)   2   5763  2881.6   108.5 <2e-16 ***
Residuals            147   3905    26.6
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Requirements for ANOVA

ANOVA requires some assumptions to be met:

1. Each treatment/population's **response variable is normally distributed**
2. The **variance (s^2) of the response variables is the same** for all treatments/populations
3. The **observations are independent**: the response variables are not related between groups

But it is fairly robust to minor violations

ANOVA is robust to ***minor violations*** *of* ***normality*** *and* ***variances***

ANOVA is more robust with ***more observations*** *and fairly* ***equal group sizes***

But what should we do if we must greatly violate these assumptions?

Kruskal Wallis Test of Multiple Samples

An extension of Mann-Whitney/Wilcox for multiple groups!

$\mathbf{H_{null}}$: All groups would give you similar a value if randomly drawn from them

$\mathbf{H_{alt}}$: At least one group would give you a larger value than another if randomly drawn

	strategy	sales	ranks
1	t1	70	136.5
2	t1	56	40.0
3	t1	69	131.0
4	t1	66	113.5
5	t1	65	104.5
.	.	.	.
51	t2	66	113.5
52	t2	64	94.5
53	t2	64	94.5
54	t2	68	125.0
55	t2	68	125.0
.	.	.	.
101	t3	50	17.0
102	t3	47	11.0
103	t3	50	17.0
104	t3	54	30.5
105	t3	50	17.0
.	.	.	.

*These **rank sums** ($\boldsymbol{R_i}$) give us a sense of how big the values are in each group*

1. **Rank all the combined values** across groups *(i=3)*

Use the `rank()` function in R

```
sales_ranks <- rank(. . .)
```

```
 [1] 136.5  40.0 131.0 113.5 104.5  86.5 113.5  48.5  94.5  86.5 136.5  71.5 104.5  56.0
[15] 131.0  56.0  94.5  65.0  56.0 131.0  30.5  30.5 113.5 104.5  94.5  71.5  36.0 125.0
[29]  36.0  56.0  78.5 125.0  65.0  78.5 125.0  65.0  86.5 104.5  56.0  86.5  48.5  86.5
[43] 104.5  94.5  48.5  94.5  56.0  43.5  78.5  78.5 113.5  94.5  94.5 125.0 125.0  48.5
[57] 104.5 104.5 131.0  94.5 149.0 149.0  78.5 140.5 131.0 113.5 104.5  94.5 140.5 144.5
[71]  86.5 113.5 104.5 147.0 120.0  65.0  78.5 131.0 136.5 120.0 140.5 113.5 120.0  65.0
[85]  71.5 144.5 113.5 120.0 104.5 136.5 144.5 144.5 120.0 140.5 131.0 149.0  65.0  94.5
[99]  78.5  65.0  17.0  11.0  17.0  30.5  17.0  30.5  21.5   9.5  21.5  65.0   6.0  26.5
[113]   7.0  56.0   9.5  36.0   5.0   3.5  21.5  56.0  21.5  78.5  56.0  30.5  43.5   1.0
[127]  40.0  40.0  12.5  12.5  71.5  14.0   3.5  30.5  17.0  24.5   2.0  36.0   8.0  26.5
[141]  17.0  65.0  24.5  48.5  36.0  43.5  78.5  43.5  78.5  48.5
```

2. **Group the ranks** into original groups

```
group_ranks <- split(. . .)
```

```
$t1
 [1] 136.5  40.0 131.0 113.5 104.5  86.5 113.5  48.5  94.5  86.5 136.5  71.5 104.5  56.0
[15] 131.0  56.0  94.5  65.0  56.0 131.0  30.5  30.5 113.5 104.5  94.5  71.5  36.0 125.0
[29]  36.0  56.0  78.5 125.0  65.0  78.5 125.0  65.0  86.5 104.5  56.0  86.5  48.5  86.5
[43] 104.5  94.5  48.5  94.5  56.0  43.5  78.5  78.5

$t2
 [1] 113.5  94.5  94.5 125.0 125.0  48.5 104.5 104.5 131.0  94.5 149.0 149.0  78.5 140.5
[15] 131.0 113.5 104.5  94.5 140.5 144.5  86.5 113.5 104.5 147.0 120.0  65.0  78.5 131.0
[29] 136.5 120.0 140.5 113.5 120.0  65.0  71.5 144.5 113.5 120.0 104.5 136.5 144.5 144.5
[43] 120.0 140.5 131.0 149.0  65.0  94.5  78.5  65.0

$t3
 [1] 17.0 11.0 17.0 30.5 17.0 30.5 21.5  9.5 21.5 65.0  6.0 26.5  7.0 56.0  9.5 36.0  5.0
[18]  3.5 21.5 56.0 21.5 78.5 56.0 30.5 43.5  1.0 40.0 40.0 12.5 12.5 71.5 14.0  3.5 30.5
[35] 17.0 24.5  2.0 36.0  8.0 26.5 17.0 65.0 24.5 48.5 36.0 43.5 78.5 43.5 78.5 48.5
```

3. **Sum the ranks** for each group

```
sapply(group_ranks, sum)
```

```
    t1      t2      t3
4159.0 5645.5 1520.5
```

*But these rank sums also depend on the **size of each group** ($\boldsymbol{n_i}$)*

4. Apply the **Kruskal Wallis formula** to sum the squared ranks

There are multiple ways of computing H
We will look at the rank sum approach…

$$Kruskal\ Wallis\ H = \frac{12}{N(N+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(N+1)$$

R_i – *sum of ranks of group i*

n_i – *number of observations in group i*

We adjust the rank sums by their group size, and compute a value that follows a chi-square distribution

```
H   # [1] 92.49268
```

5. **Find p-value** of H from χ^2 distribution

p-value tells you significance of test

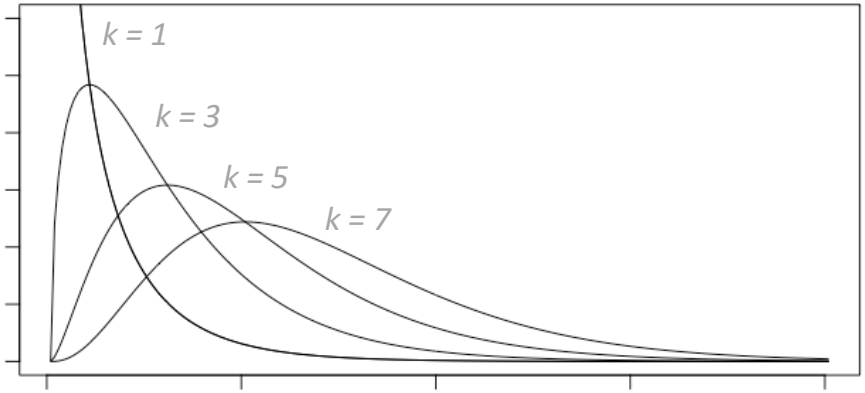
```
kw_p <- 1 - pchisq(H, df=k-1)
```

k – *number of groups*

```
kw_p
[1] 0
```

Reject H_{null}

χ^2 "Chi-square" distribution



R has a built-in Kruskal-Wallis Test!

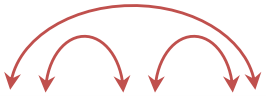
```
kruskal.test(sales ~ strategy, data = ads)

Kruskal-Wallis chi-squared = 92.679, df = 2, p-value < 2.2e-16
```

Note: *There will be some difference in the calculation of H because* `kruskal.test()` *accounts for ties in ranks…*

Multigroup *Post hoc* Tests

After data or results are seen



	t1	t2	t3
1	70	66	50
2	56	64	47
3	69	64	50
4	66	68	54

There are three possible group-wise differences:
t1-vs-t2, t1-vs-t3, t2-vs-t3

So which groups are different from which others?

It is *tempting* to use multiple two-Sample Tests Post-hoc

ANOVA  **3 t-tests**

```
t.test(adverts$t1, adverts$t2)
t.test(adverts$t1, adverts$t3)
t.test(adverts$t2, adverts$t3)
```

But do not use two-sample methods to test two-way differences between multiple groups!

Kruskal-Wallis  **3 Wilcoxon Tests**

```
wilcox.test(adverts$t1, adverts$t2)
wilcox.test(adverts$t1, adverts$t3)
wilcox.test(adverts$t2, adverts$t3)
```

Family-Wise Error

single dice

Running a test at 5% significance (α=0.05) is like rolling a 20-sided dice

Imagine rolling **one** 20-sided dice; probability of getting a "1" each time is 5%

```
rbinom(n=1, size=1, prob=0.05)    0, 0, 0, 0, 0, 0, 0, 0, 0, 0
```

This is the **single** risk of Type I error at 5% significance

```
sum(replicate(n, sum(rbinom(n=1, size=1, prob=0.05) >= 1))) / n
[1] 0.05055
```

family of dice

Imagine rolling **five** 20-sided dice; probability of getting a "1" each time is **22**%

```
rbinom(n=1, size=5, prob=0.05)    0, 0, 1, 0, 0, 0, 0, 0, 1, 0
```

This is the **family-wise** risk of type I error at 5% significance

```
sum(replicate(n, sum(rbinom(n=1, size=5, prob=0.05) >= 1))) / n
[1] 0.22659
```

☹

When evaluating many hypotheses, you are quite likely to commit type I error

→

☹

So we cannot assess a family of tests using the same method to assess a single test

Tukey Test: Post-Hoc Test for ANOVA

Confidence interval of difference of means

$\hat{\sigma}_\varepsilon$ – *standard error of ANOVA* ($\sqrt{MSE}$)

difference of group means

$$\bar{y}_{i\cdot} - \bar{y}_{j\cdot} \pm \frac{1}{\sqrt{2}} q_{\alpha;\, r,\, N-r}\ \hat{\sigma}_\epsilon \sqrt{\frac{2}{n}}$$

n – *size of a single group*

q – *adjustment factor based on number of groups*

```
TukeyHSD(anova_model, conf.level = 0.01)
```

	diff	lwr	upr	p adj
t2-t1	4.38	4.240815	4.519185	**0.000112**
t3-t1	-10.40	-10.539185	-10.260815	**0.000000**
t3-t2	-14.78	-14.919185	-14.640815	**0.000000**

Considers number of groups
Considers unexplained variance (MSE)

Assumptions of Tukey Test

1. The **variance (s^2) of the response variables is the same** for all treatments/populations
2. The **observations are independent**: the response variables are not related between groups

Dunn Test: Posthoc Tests for Kruskal Wallis

difference of mean group sum-ranks

$$Z_{kw} = \frac{\overline{R}_i - \overline{R}_j}{\sqrt{\frac{N(N+1)}{12}\left(\frac{1}{n_i} + \frac{1}{n_j}\right)}}$$

$\overline{R}_i$, $\overline{R}_j$ – *mean of sum of ranks for groups i and j*

N – *total number of data observations*

n_i , n_j – *size of groups i and j*

```
# install.packages("FSA")
library(FSA)
dunnTest(sales ~ strategy, data = ads, method = "bonferroni")
```

```
  Comparison         Z      P.unadj        P.adj
1    t1 - t2 -3.42499 6.148226e-04 1.844468e-03
2    t1 - t3  6.07927 1.207306e-09 3.621918e-09
3    t2 - t3  9.50426 2.014750e-21 6.044251e-21
```

No assumptions *of variance homogeneity*
No assumptions *of equal group sizes*

Parametric vs. Non-Parametric Methods

	Parametric Methods	**Non-parametric Methods**
Data Distribution Assumptions	*Distribution parameters* of original data	--
Approach	*Statistical* (simple, elegant formulas)	*Computational* (resampling/ranking)
Advantage	More statistical power if assumptions are met	Better when assumptions violated; Useful for new/different statistics
Disadvantage	Often not applicable to noisy real-world data; Often not applicable to new/different statistics	Procedures get complicated quickly; Many variations to choose from

	Parametric Tests	Non-parametric Tests
Assumptions that need to be met	*Distribution parameters* of original data	--
Information that is tested	*Mean, Variance*	*Median, Quantiles, Ranks, Deviation, …*
Example of **One-sample test**	t-Tests	Bootstrap
Example of **Two-sample test**	t-Tests	Bootstrapped t-Tests Parametric Test Wilcoxon Test Signed-Rank Test
Example of **Multi-group test**	ANOVA	Bootstrapped ANOVA Kruskal Wallis Test
Posthoc multigroup comparisons	Tukey Test	Dunn Test
Major Disadvantage	Assumptions often cannot be met in real world data	Gets very complicated quickly; Many variations and alternatives