ABI Summer 2021 Introduction to Statistics

Guest Session 3

Javier Cabrera*, Volha Tryputsen** & Davit Sargsyan**

July 13, 2021

(*)Rutgers, the State University of New Jersey
(**)Janssen Pharmaceutical Research and Development (J&J)

Group Assignment

Group 1	Group 2	Group 3	Group 4	Group 5
Gisane (R)	Arabo (R)	Susie (R)	Tamara (R)	Mher (R)
Arvin	Nerses	Ashkhen	Vardan	Vika
Satenik	Liana	Anush (R)	Hripsime	Nelli

The Royal Guinea Pig Problem

- The Guinea pig lost appetite and began losing weight . It even stopped eating its favorite treat - carrot

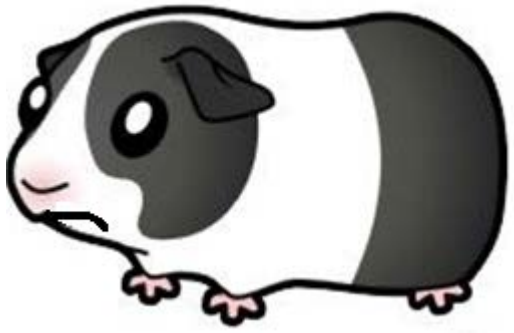
- You are commissioned to find a cure for Peppa – and get half of the Kingdom

- Or else…

Available resources:

- Castle basement full of mice and rats
- 2 types of diet: regular grass or carrot-rich
- 3 natural compounds to use as treatments

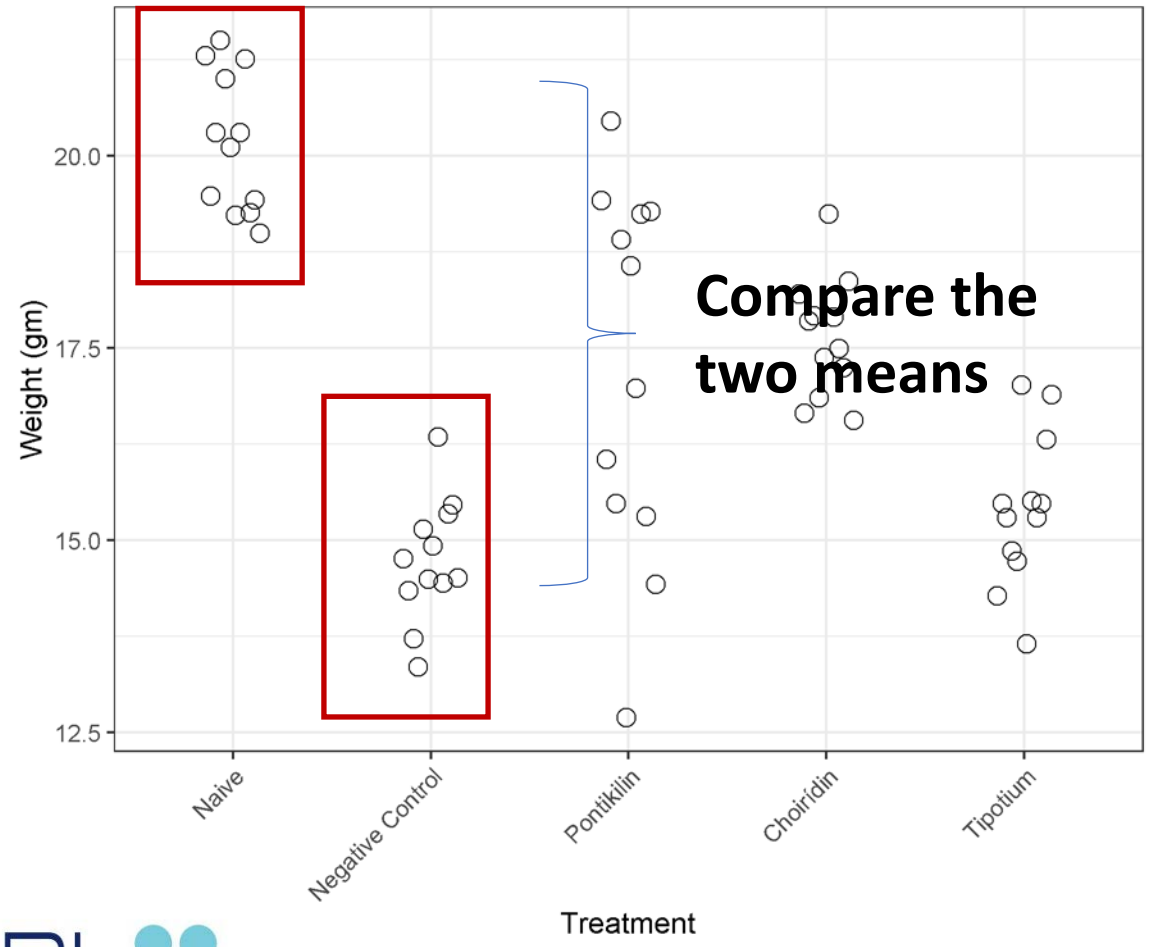
Peppa's Information
Age: 5 years 2 months; Sex: Female, Skin Color: Pink;
Favorite snack: carrots



	Last Week	This Week
Diet	Grass and Carrots	Grass
Weight	550g	423g
Food Intake	128g/day	105g/day

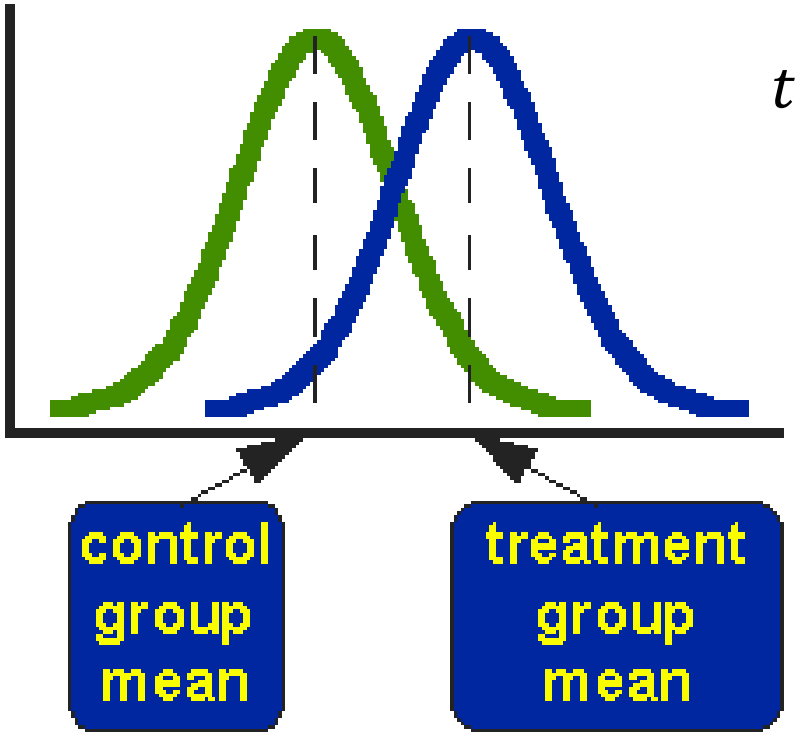
The Royal Guinea Pig Problem

Compare means: t-Test



- First, we want to make sure that our experimental (animal) model worked: there is a significant difference between the healthy and the sick animals
- Do a t-Test

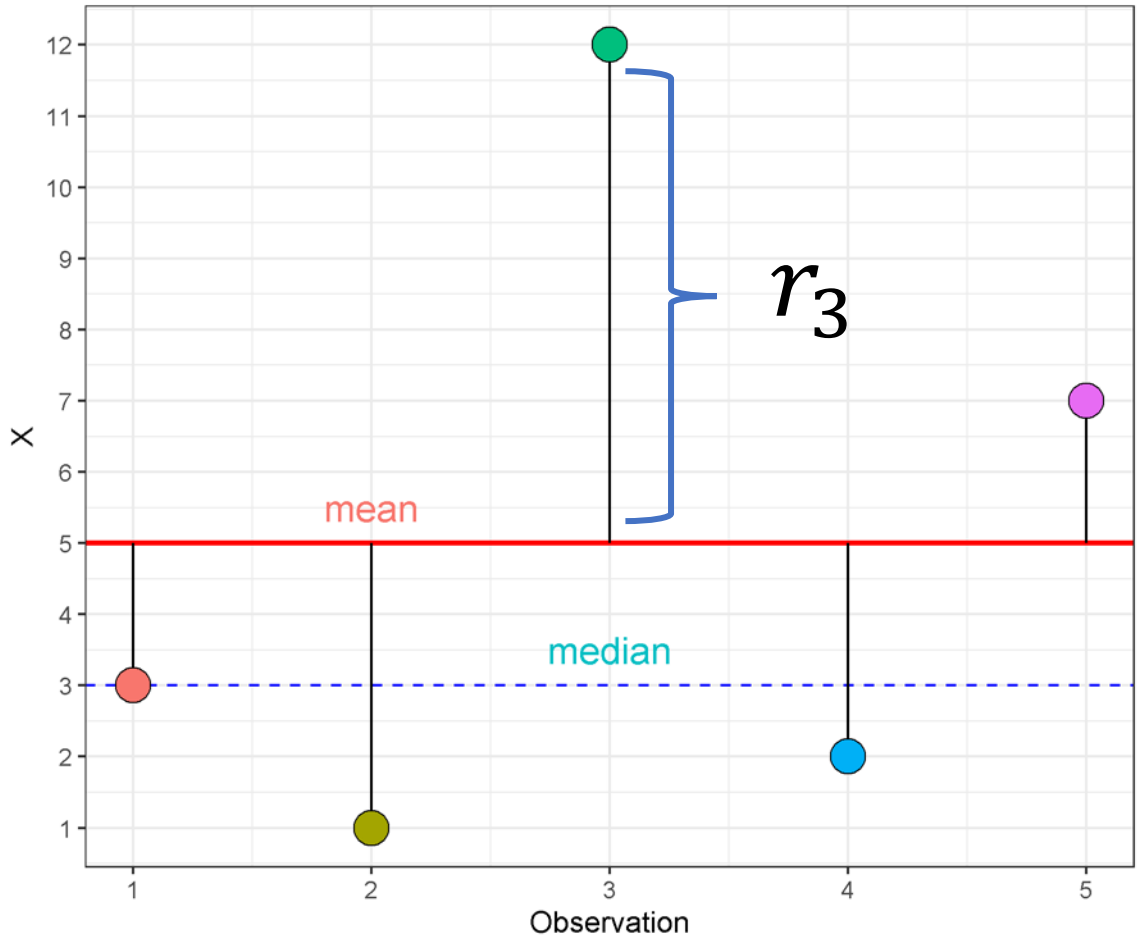
Comparing 2 Groups' Means: t-Test



$$t = \frac{signal}{noise} = \frac{difference\ between\ two\ means}{pooled\ standard\ error\ of\ the\ means}$$

$$t = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{\frac{Var_A}{n_A} + \frac{Var_B}{n_B}}}$$

Median, Mean and Variance



Obs.	Value
2	1
4	2
1	3
5	7
3	12

Mean (or average) = sum of observation values/ number of observations

$$\bar{x} = {\sum_{i=1}^{N} x_i}/{N}$$

Residual (or error) = observation value – mean

$$r_i = x_i - \bar{x}$$

Residual sum of squares :

$$RSS = \sum_{i=1}^{N} r_i^2$$

Standard deviation:

$$Sample\ Variance = \frac{RSS}{N-1}$$

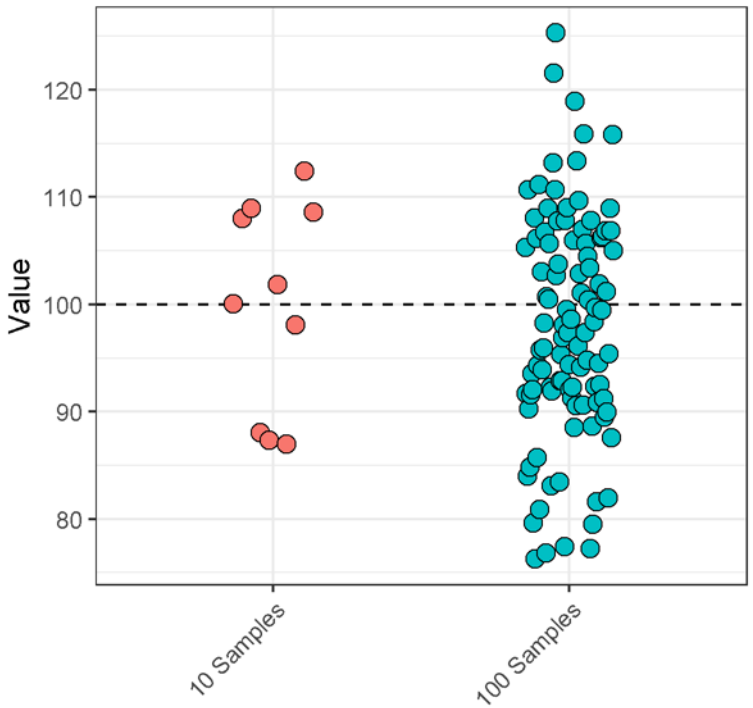
Source: https://www.123rf.com/photo_42732599_a-cartoon-queen-with-a-confused-expression-.html

Standard Deviation vs. Standard Error

Standard deviation describes **variability** in the data

$$SD = \sqrt{Variance}$$

Population mean = 100; SD = 10



Standard error describes statistical **accuracy** of an estimate

$$SE = \sqrt{\frac{Variance}{N}}$$

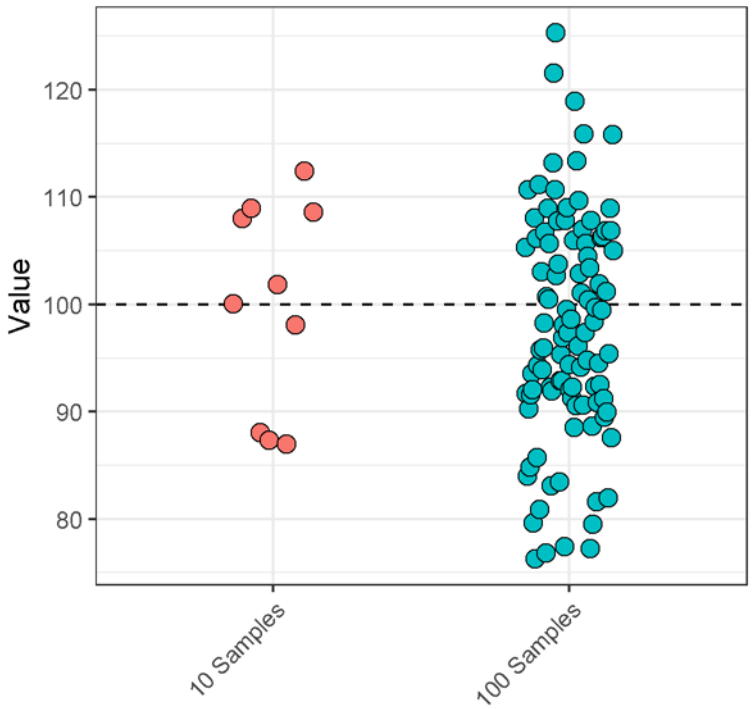
Standard Deviation vs. Standard Error

Standard deviation describes **variability** in the data

$$SD = \sqrt{Variance}$$

SD **does not** depend on the number of observations in the sample although it might be smaller in smaller samples because extreme observations are unlikely to be sampled

Population mean = 100; SD = 10



	10 samples	100 samples
SD	9.73	10.40
SEM	3.08	1.04

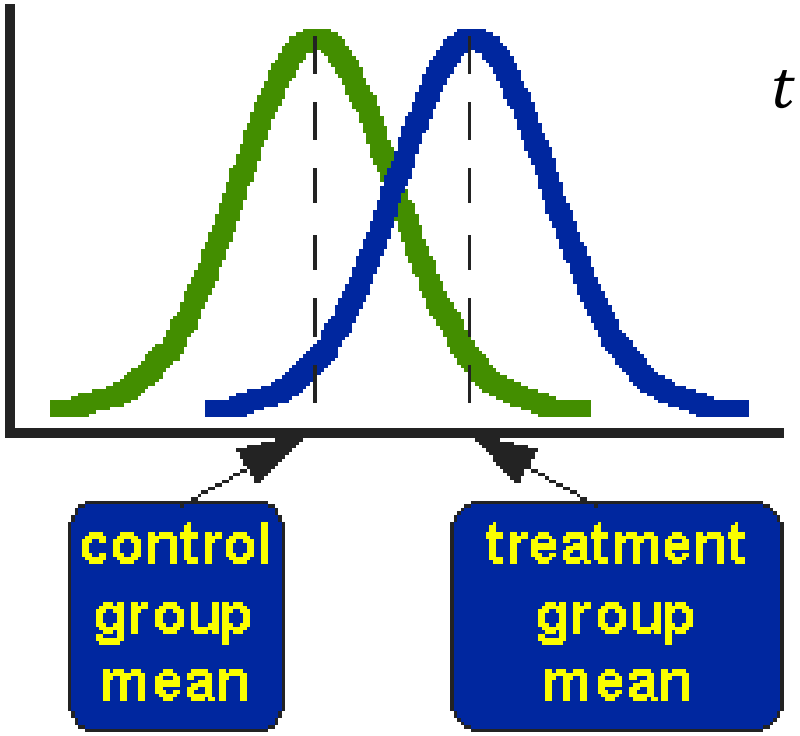
Standard error describes statistical **accuracy** of an estimate

$$SE = \sqrt{\frac{Variance}{N}}$$

SE generally **gets smaller** as sample size increases

NOTE: think about getting more evidence for your guess

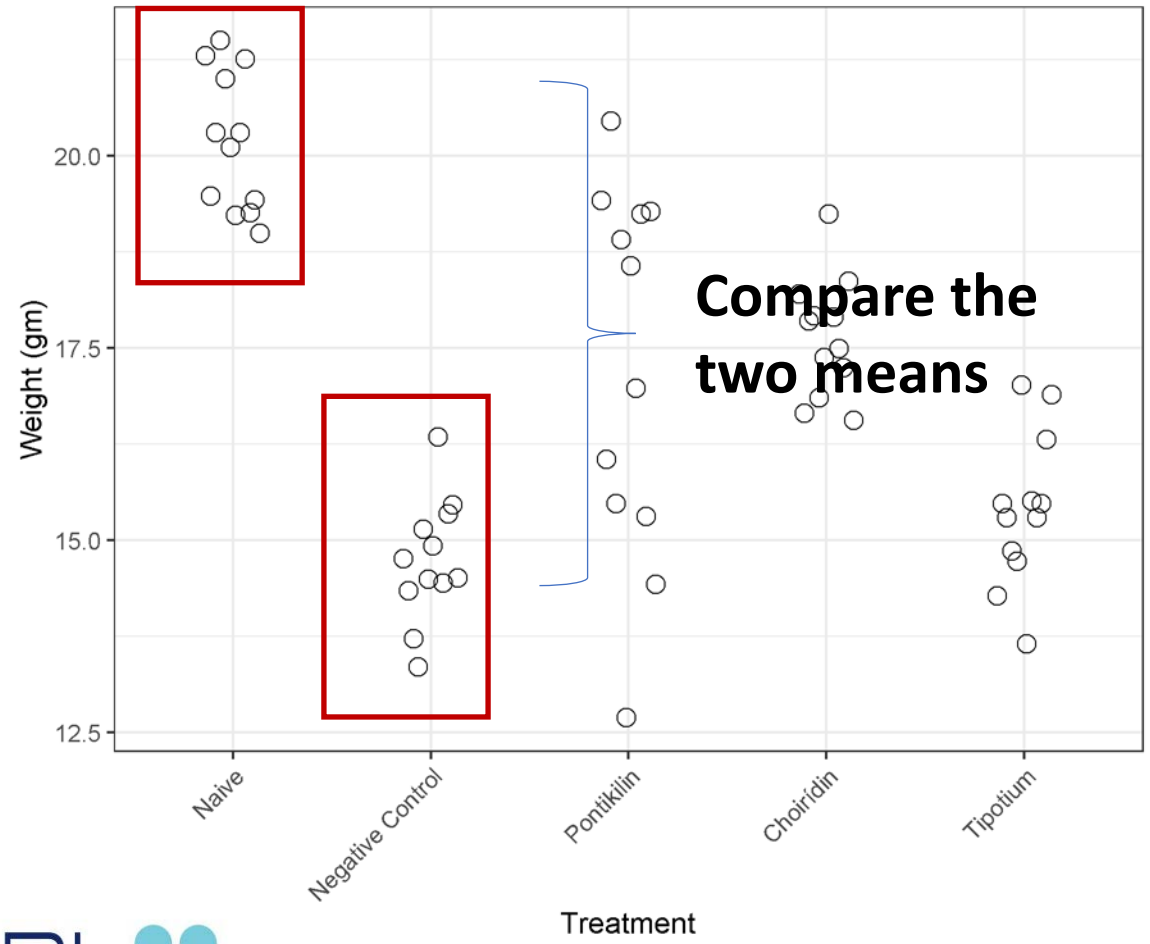
Comparing 2 Groups' Means: t-Test



$$t = \frac{signal}{noise} = \frac{difference\ between\ two\ means}{pooled\ standard\ errors\ of\ the\ means}$$

$$t = \frac{\bar{x}_A - \bar{x}_B}{\sqrt{\frac{Var_A}{n_A} + \frac{Var_B}{n_B}}}$$

The Royal Guinea Pig Problem




```
# t-Test: controls only
```{r}
t.test(Weight ~ Treatment,
       data = dt1[Treatment %in% c("Naive",
                                   "Negative Control"), ],
       var.equal = TRUE)
```
```

```
        Two Sample t-test

data:  Weight by Treatment
t = 15.611, df = 22, p-value = 2.195e-13
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 4.721633 6.168367
sample estimates:
           mean in group Naive mean in group Negative Control
                        20.180                         14.735
```

Two sample t-test assumptions

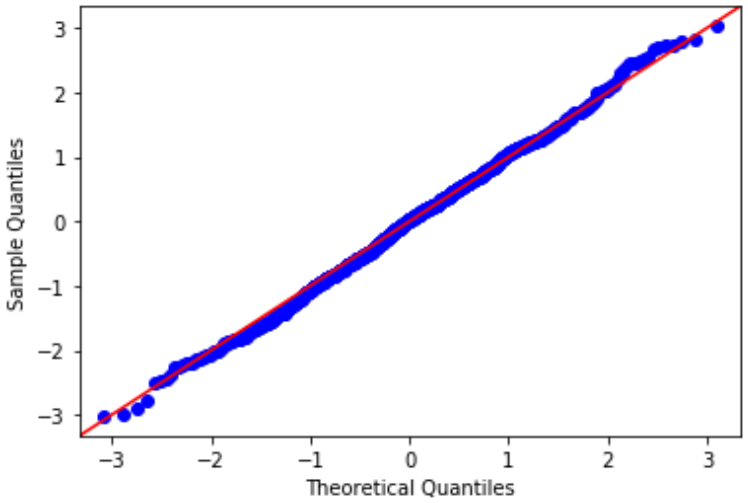
- Data values must be independent: measurements for one observation do not affect measurements for any other observation.
- Data in each group must be obtained via a random sample from the population.
- Data in each group are normally distributed.
- Data values are continuous.
- The variances for the two independent groups are equal.

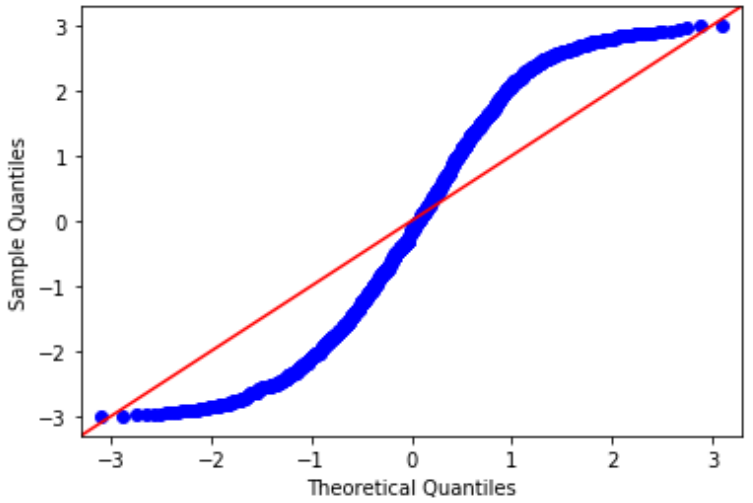
Two sample t-test assumptions: check

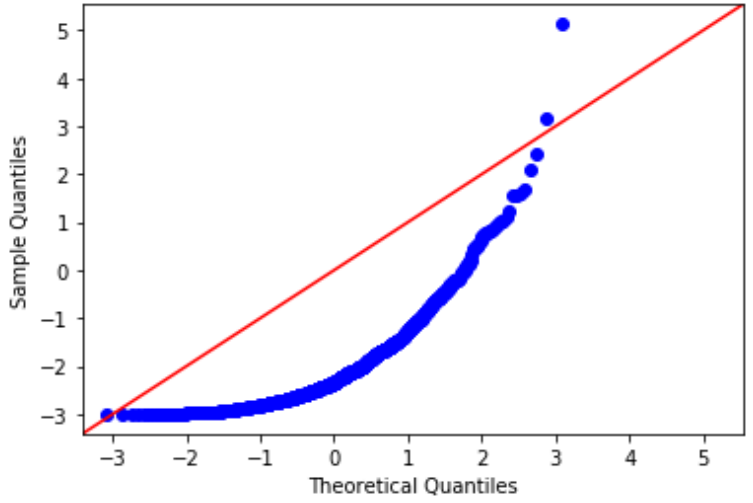
- Data values must be independent: measurements for one observation do not affect measurements for any other observation.
- Data in each group must be obtained via a random sample from the population.
- Data in each group are normally distributed.
- Data values are continuous.
- The variances for the two independent groups are equal.

Checking t-test assumptions: (1) normality

qq-plot to check for Normality

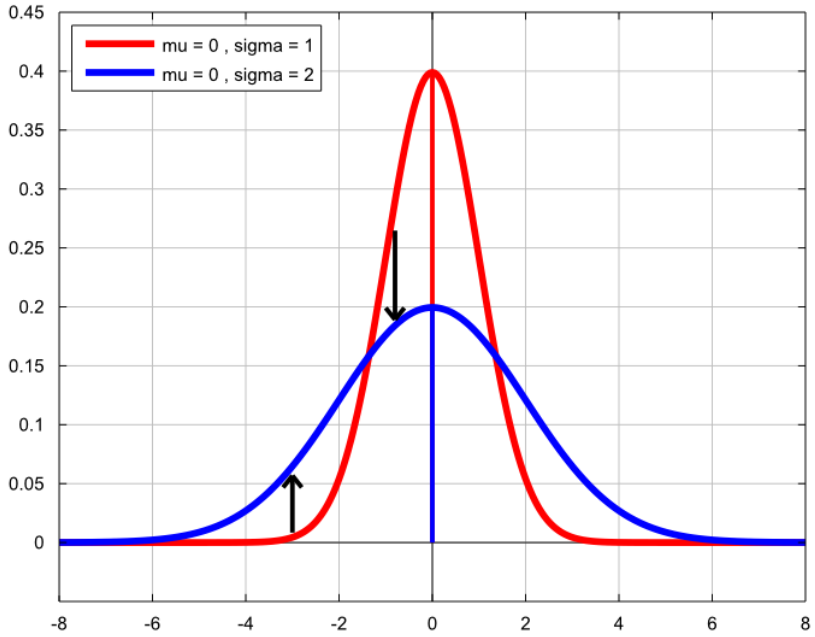


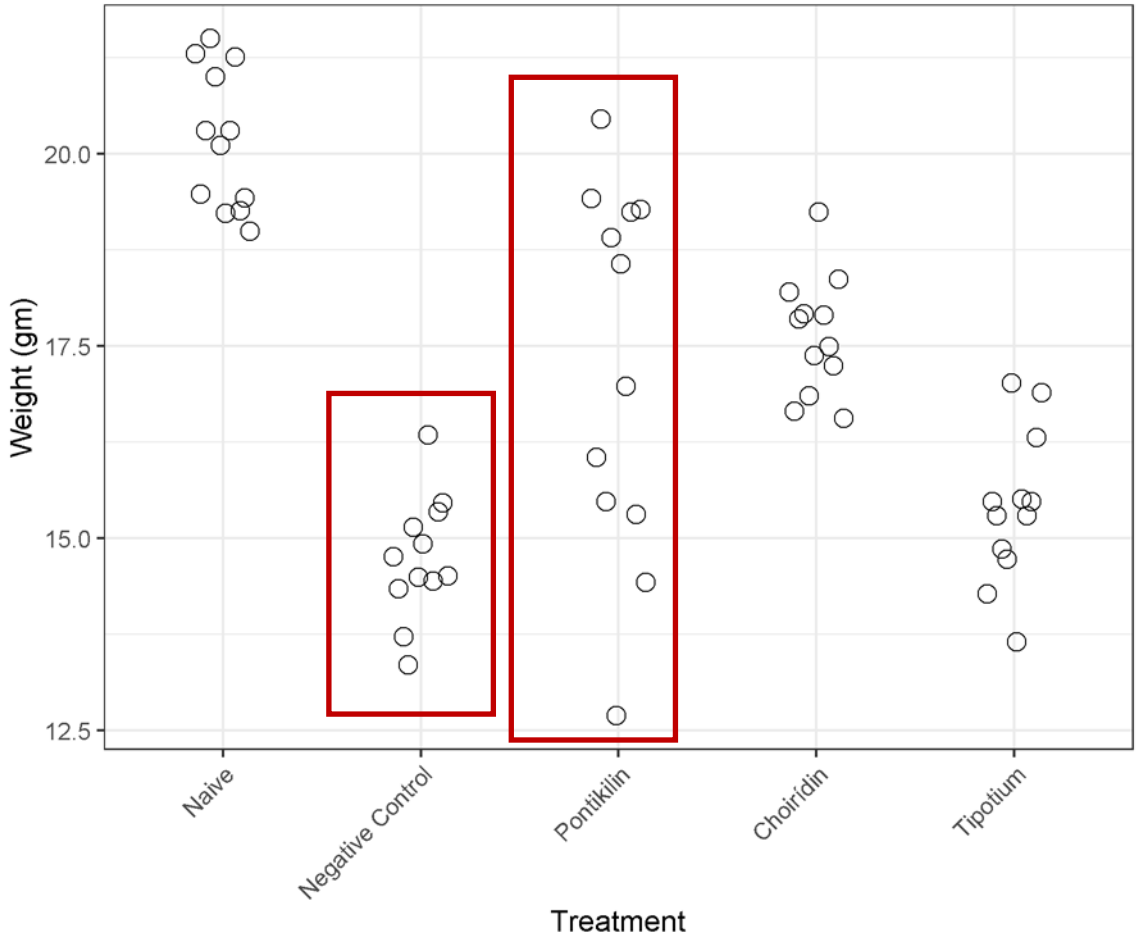




Checking t-test assumptions: (2)equal variances

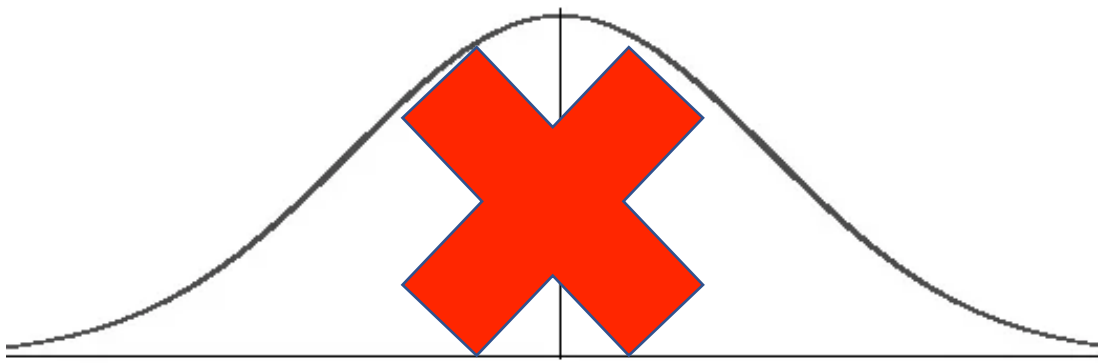
- 2 variances are NOT equal!
- Use the unequal variance t- test, also called the Welch t- test.





What if data is not Normally distributed?

Use Wilcoxon test - non-parametric (distribution-free)!



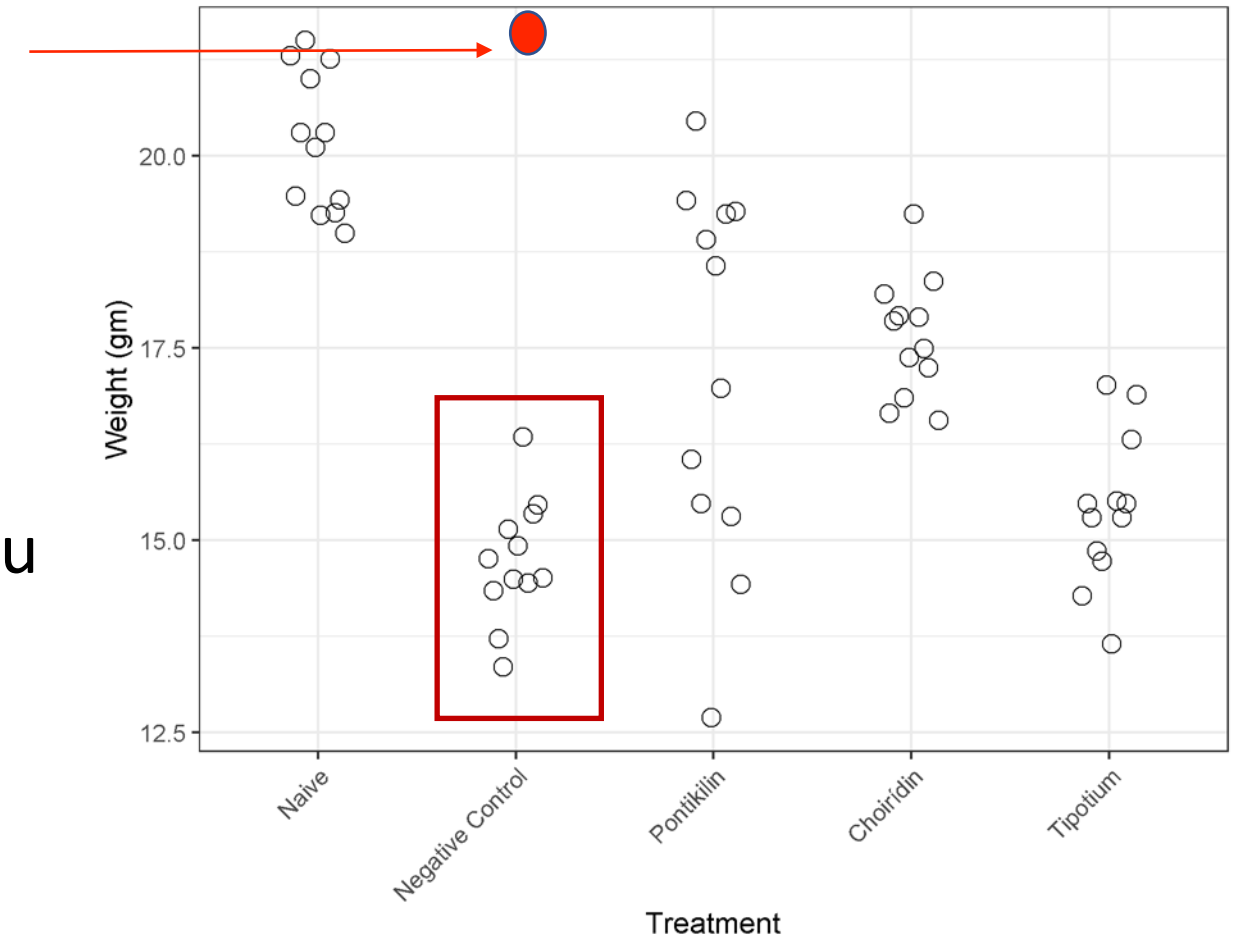
```
# Wilcoxon test
```{r}
wilcox.test(Weight ~ Treatment,
       data = dt1[Treatment %in% c("Naive",
                                   "Negative Control"), ])
```
```

```
cannot compute exact p-value with ties
        Wilcoxon rank sum test with continuity correction

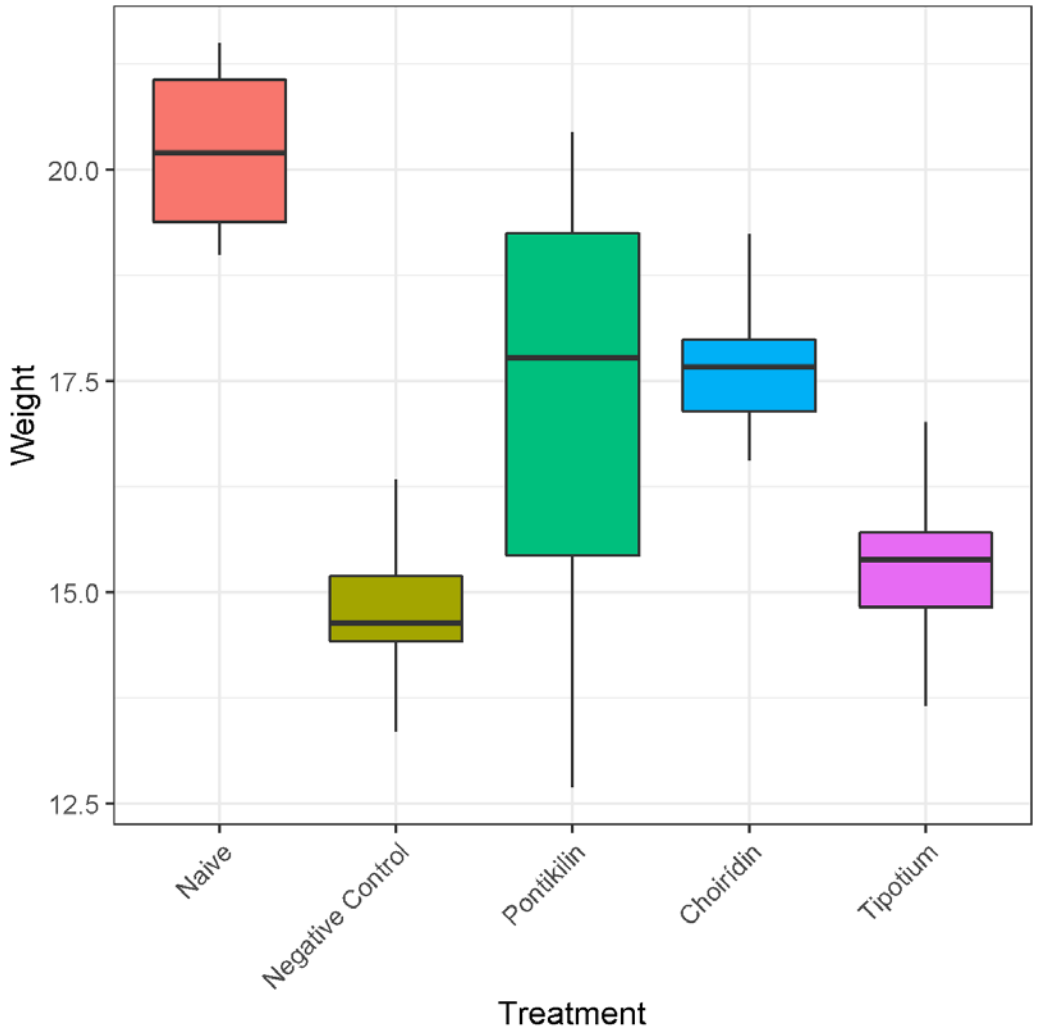
data:  Weight by Treatment
W = 144, p-value = 3.644e-05
alternative hypothesis: true location shift is not equal
to 0
```

What if data has outliers?

- An outlier is a data point that differs significantly from other observations in a group
- Use Wilcoxon test!
- If you can explain the outlier, you can remove it (needs good justification). Can create a problem (Ozone layer data example)



Boxplot




```
# Boxplot
```{r}
p2 <- ggplot(dt1,
             aes(x = Treatment,
                 y = Weight,
                 fill = Treatment)) +
  geom_boxplot() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45,
                                   hjust = 1),
        legend.position = "none")
p2
```
```

Analysis of Variance (ANOVA)

```
# ANOVA
```{r}
m1 <- aov(Weight ~ Treatment,
          data = dt1)
summary(m1)
```
```

```
             Df Sum Sq Mean Sq F value   Pr(>F)
Treatment     4 219.12   54.78    30.7 1.94e-13 ***
Residuals    55  98.14    1.78
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

POST HOC Pairwise Comparison

```
# Pairwise comparison
```{r}
# p-values adjusted for multiplicity
m1_mult_comp_adj <- glht(m1,
                         linfct = mcp("Treatment" = "Tukey"))
m1_mult_comp_adj
summary(m1_mult_comp_adj)
```
```

POST HOC Pairwise Comparison

```
         Simultaneous Tests for General Linear Hypotheses

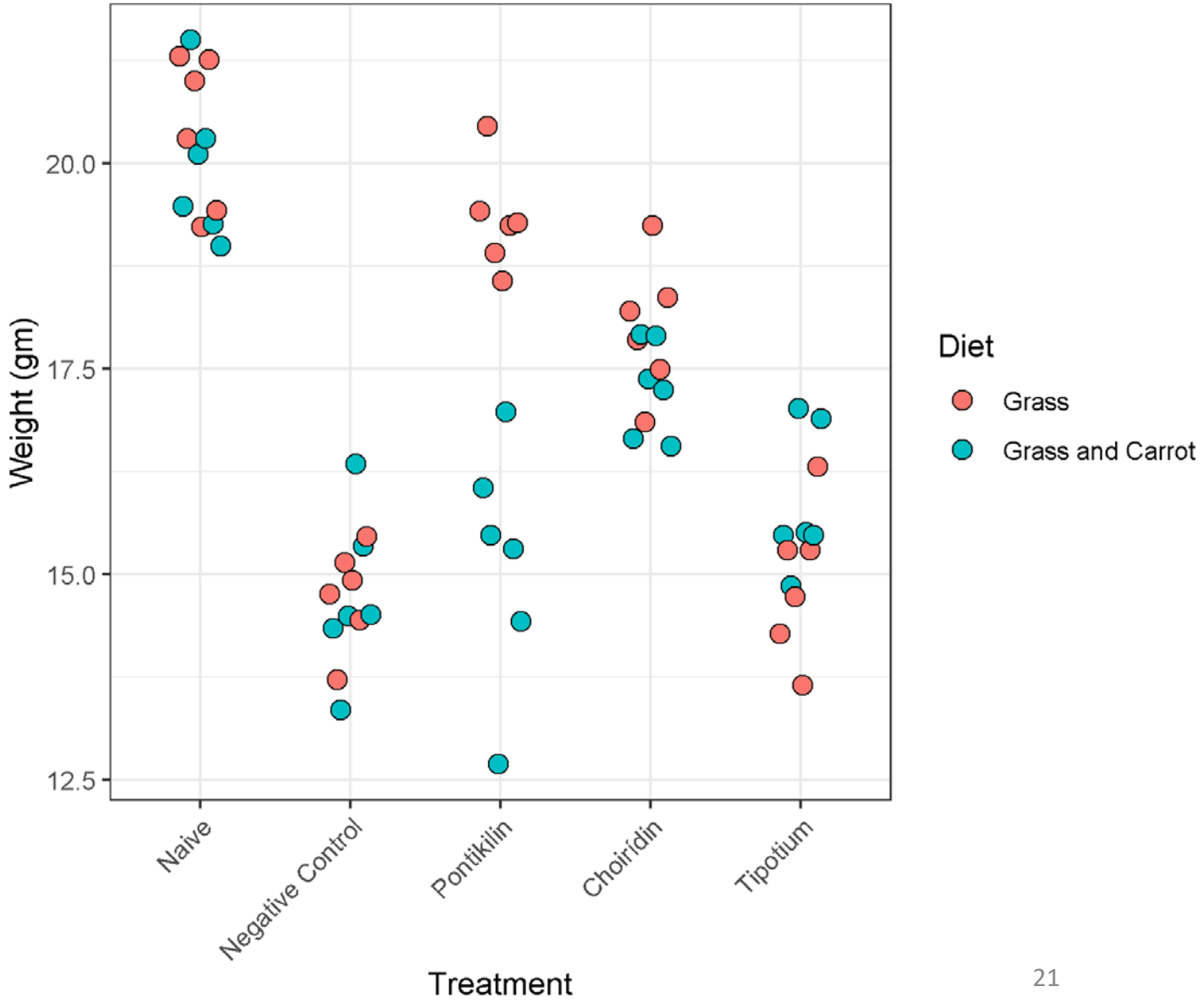
Multiple Comparisons of Means: Tukey Contrasts


Fit: aov(formula = Weight ~ Treatment, data = dt1)

Linear Hypotheses:
                                    Estimate Std. Error t value Pr(>|t|)
Negative Control - Naive == 0       -5.4450     0.5453  -9.985  < 1e-04 ***
Pontikilin - Naive == 0             -2.9458     0.5453  -5.402  < 1e-04 ***
Choirídin - Naive == 0              -2.5425     0.5453  -4.662 0.000199 ***
Tipotium - Naive == 0               -4.7808     0.5453  -8.767  < 1e-04 ***
Pontikilin - Negative Control == 0   2.4992     0.5453   4.583 0.000246 ***
Choirídin - Negative Control == 0    2.9025     0.5453   5.322  < 1e-04 ***
Tipotium - Negative Control == 0     0.6642     0.5453   1.218 0.741052
Choirídin - Pontikilin == 0          0.4033     0.5453   0.740 0.946183
Tipotium - Pontikilin == 0          -1.8350     0.5453  -3.365 0.011774 *
Tipotium - Choirídin == 0           -2.2383     0.5453  -4.105 0.001270 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Adjusted p values reported -- single-step method)
```

What About the Diet?

- Let’s do this one together (class exercise)
- Think about how diet affects the outcome (weight)
- How does diet interact with the treatment?
- How can we analyze diet effect ?



What Are Our Recommendations?

- Is there any treatment that might prevent weight loss in the guinea pig?
- Are some treatments better than others? Why and why not?
- What limitations did your experiment have?
- What would you do differently if you had more time and recourses?

The Royal Guinea Pig Problem

And they lived happily ever after…

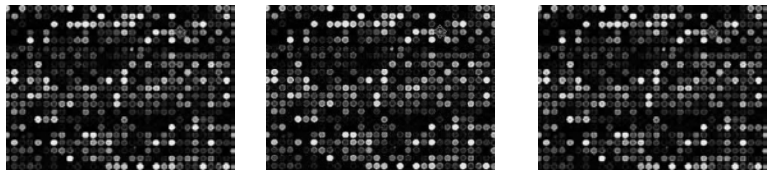
*Because **You** found the cure!*

Source: https://www.123rf.com/photo_42732599_a-cartoon-queen-with-a-confused-expression-.html

Case study

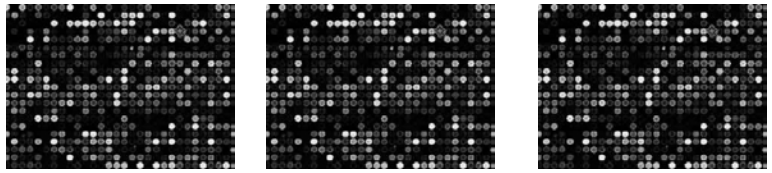
♦ **Experiment:** Compare the gene expression profiles of 3 KO mice vs 3 WT mice using a microarray with 22283 genes.

WT:



C1 C2 C3

KO:



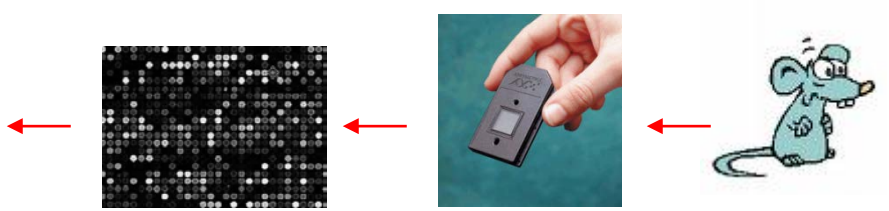
T1 T2 T3

Hereafter: variables = genes

Gene expression matrix

- **Data:** Expression measures for *G* genes in *N* samples:

	C1	C2	C3	T1	T2	T3
G1	83	94	82	111	130	122
G2	16	14	7	2	11	33
G3	490	879	193	604	1031	962
G4	46458	49268	74059	44849	42235	44611
G5	32	70	185	20	25	19
G6	1067	891	546	906	1038	1098
G7	118	111	95	896	536	695
G8	10	30	25	24	31	28
G9	166	132	162	27	109	213
G10	136	139	44	62	23	135
	.	.	.	.	.	.

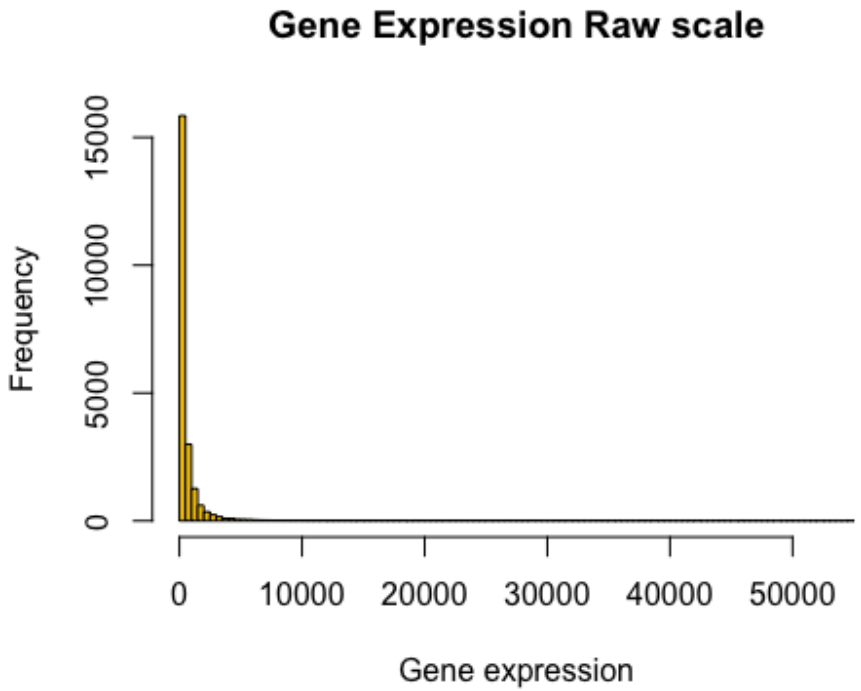


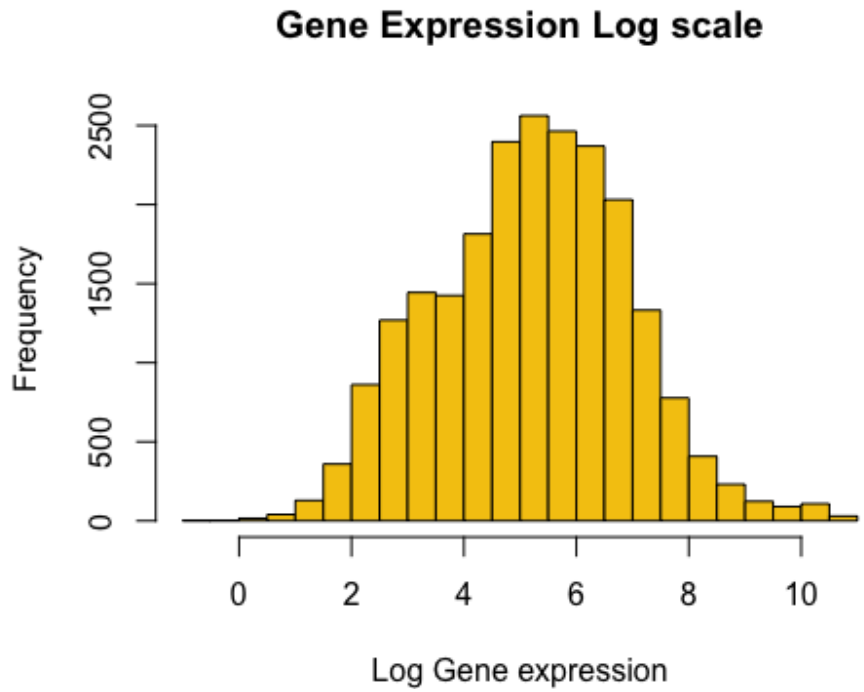
Preprocess: normalize and log transform

22283 rows (genes) x 6 columns (samples)

	C1	C2	C3	T1	T2	T3
G8521	6.89	7.18	6.60	7.40	7.15	7.40
G8522	6.78	6.55	6.37	6.89	6.78	6.92
G8523	6.52	6.61	6.72	6.51	6.59	6.46
G8524	5.67	5.69	5.88	7.43	7.16	7.31
G8525	5.64	5.91	5.61	7.41	7.49	7.41
G8526	4.63	4.85	5.72	5.71	5.47	5.79
G8527	8.28	7.88	7.84	8.12	7.99	7.97
G8528	7.81	7.58	7.24	7.79	7.38	8.60
G8529	4.26	4.20	4.82	3.11	4.94	3.08
G8530	7.36	7.45	7.31	7.46	7.53	7.35
G8531	5.30	5.36	5.70	5.41	5.73	5.77
G8532	5.84	5.48	5.93	5.84	5.73	5.73
G8533	9.45	9.56	9.92	10.15	9.81	9.36
G8534	7.57	7.55	7.30	7.48	7.82	7.46

25





It is better to log transform!

For each row -> Compute t-statistic and p-value

$$t = \frac{\bar{x}_1 - \bar{x}_2}{s_p\sqrt{(1/n_1) + (1/n_2)}},$$

Bonferroni method: Control for multiplicity

Top 10 genes (sorted by *t*-test *p*-value)

Gene	Fold	Dir	p	p(Bonf)
G6546	2.36	D	0.000004	0.0964
G19945	3.25	U	0.000005	0.1102
G21586	1.64	U	0.000008	0.1765
G18970	2.52	U	0.000019	0.4220
G7432	3.70	D	0.000033	0.7248
G19057	1.85	U	0.000046	1.0000
G17361	4.34	D	0.000067	1.0000
G8525	5.57	D	0.000067	1.0000
G425	18.11	D	0.000078	1.0000
G8524	4.74	D	0.000109	1.0000

Bonferroni

$p^{BON}_i \leftarrow \min(\#\ of\ tests \times p_i, 1)$

Or $\alpha^{BON} \leftarrow \alpha\ /\ \#\ of\ tests$

Drawbacks:

- Too restrictive.
- Too many false negatives.

Bonferroni cut off for $\alpha = 0.05$:

$\alpha^{BON} \leftarrow 0.0000023$

How to make improvements

False discoveries correction:

Put p-values in order

0.000004 0.000005 0.000008

Multiply by *# of tests* up to that one

22211 22210 22209

Bonferroni method: Control for multiplicity

Bonferroni

$p^{\text{BON}}_i \leftarrow \min(\#\ of\ tests \times p_i, 1)$

Or $\alpha^{\text{BON}} \leftarrow \alpha\ /\ \#\ of\ tests$

Drawbacks:

- Too restrictive.
- Too many false negatives.

Bonferroni cut off for $\alpha = 0.05$:

$\alpha^{\text{BON}} \leftarrow 0.0000023$

How to make improvements

False discoveries correction:

Put p-values in order

0.000004 0.000005 0.000008

Multiply by *# of tests* up to that one

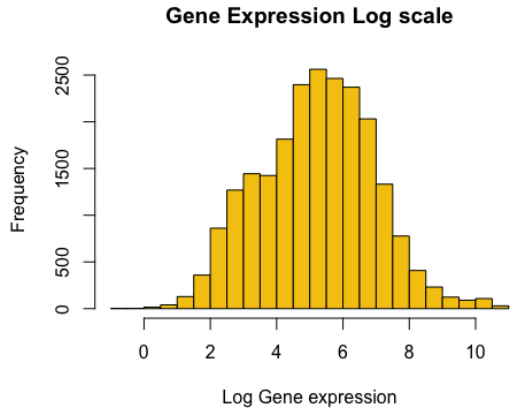
22211 22210 22209

Top 10 genes (sorted by *t*-test *p*-value)

Gene	Fold	Dir	p	p(Bonf)
G6546	2.36	D	0.000004	0.0964
G19945	3.25	U	0.000005	0.1102
G21586	1.64	U	0.000008	0.1765
G18970	2.52	U	0.000019	0.4220
G7432	3.70	D	0.000033	0.7248
G19057	1.85	U	0.000046	1.0000
G17361	4.34	D	0.000067	1.0000
G8525	5.57	D	0.000067	1.0000
G425	18.11	D	0.000078	1.0000
G8524	4.74	D	0.000109	1.0000

T-tests for microarrays or RNAseq

```
##  find this library at https://sites.rutgers.edu/javier-cabrera/research/
library(DNAMR)
data(mice.A)
par(mfrow=c(1,2))
hist(mice.Ar[,1],100,col=7,xlab="Gene expression",main="Gene Expression Raw scale")
hist(log(mice.Ar[,1]),20,col=7,xlab="Log Gene expression",main="Gene Expression Log scale")
```




```
data(mice.A)
data= as.matrix(mice.A)
tt= rttest(data[,1:3],data[,4:6])
head(round(tt,3))
```

```
          T    DF    PV
[1,]  0.600 3.993 0.581
[2,] -0.412 3.888 0.702
[3,] -1.701 2.685 0.198
[4,] -0.649 3.327 0.559
[5,] -0.120 2.964 0.912
[6,]  1.572 3.969 0.192
```

```
##  Top p-values

head(round(tt[sort.list(tt[,3]),],6))
```

```
              T       DF       PV
[1,] -34.26634 3.862807 0.000006
[2,] -20.64362 3.870865 0.000042
[3,] -17.22758 3.890077 0.000081
[4,]  33.13688 2.806082 0.000100
[5,] -16.57884 3.748933 0.000120
[6,] -15.19964 3.901898 0.000129
```

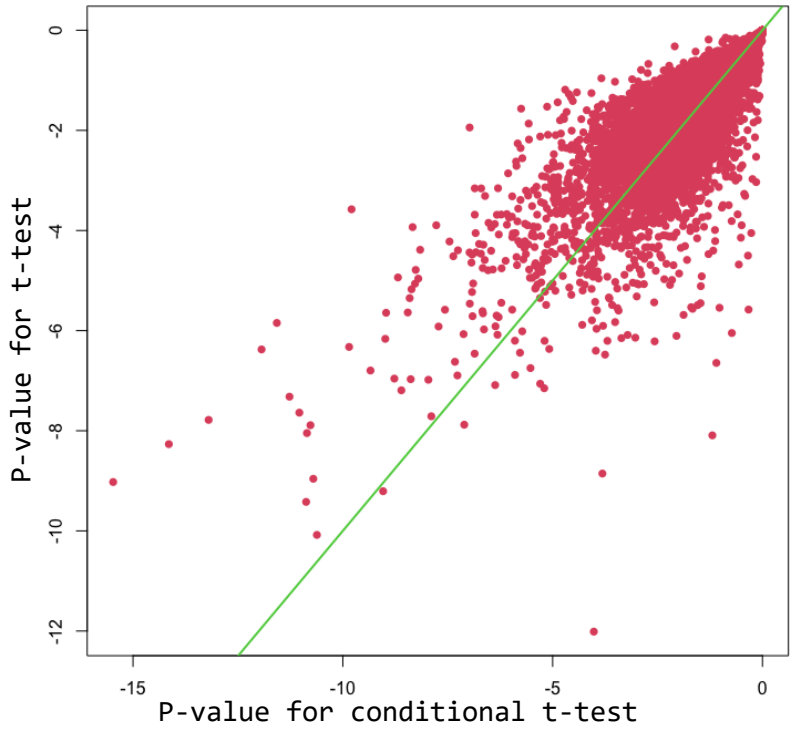
T-tests p-values vs Conditional test p-values

$$t^* = \frac{\bar{x}_1 - \bar{x}_2}{s_p^* \sqrt{(1/n_1) + (1/n_2)}},$$

```
ctpv= ct2(data[,1:3],data[,4:6])

par(mfrow=c(1,1))
plot(log(ctpv),log(tt[,3]),pch=16,col=2)
abline(0,1,col=3,lwd=2)


sort(ctpv)[1:10]*22211
 [1] 0.004233567 0.015955578 0.041185283 0.145401025
0.209298907 0.282893769
 [7] 0.357705307 0.420474419 0.428882406 0.466184181
```



Similarly we developed Conditional F test and p-values
Amaratunga Cabrera (2004)
Amaratunga Cabrera & Shkedy(2014)

```
library(DNAMR)
data(Khan)
names(Khan)
names(Khan)

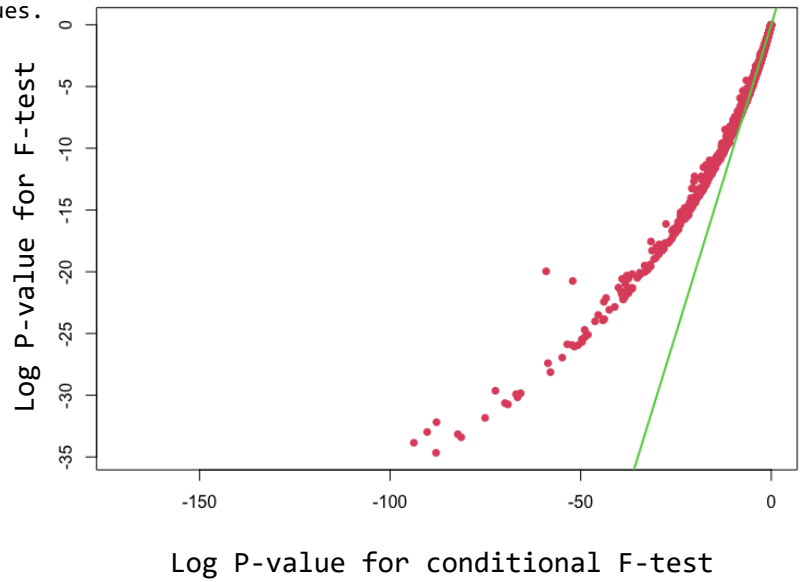
 [1] "ImageId"    "TRAIN1.EW"  "TRAIN2.EW"  "TRAIN3.EW"  "TRAIN4.EW"  "TRAIN5.EW"  "TRAIN6.EW"  "TRAIN7.EW"  "TRAIN8.EW"  "TRAIN9.EW"
[11] "TRAIN10.EW" "TRAIN11.EW" "TRAIN12.EW" "TRAIN13.EW" "TRAIN14.EW" "TRAIN15.EW" "TRAIN16.EW" "TRAIN17.EW" "TRAIN18.EW" "TRAIN19.EW"
[21] "TRAIN20.EW" "TRAIN21.EW" "TRAIN22.EW" "TRAIN23.EW" "TRAIN24.BL" "TRAIN25.BL" "TRAIN26.BL" "TRAIN27.BL" "TRAIN28.BL" "TRAIN29.BL"
[31] "TRAIN30.BL" "TRAIN31.BL" "TRAIN32.NB" "TRAIN33.NB" "TRAIN34.NB" "TRAIN35.NB" "TRAIN36.NB" "TRAIN37.NB" "TRAIN38.NB" "TRAIN39.NB"
[41] "TRAIN40.NB" "TRAIN41.NB" "TRAIN42.NB" "TRAIN43.NB" "TRAIN44.RM" "TRAIN45.RM" "TRAIN46.RM" "TRAIN47.RM" "TRAIN48.RM" "TRAIN49.RM"
[51] "TRAIN50.RM" "TRAIN51.RM" "TRAIN52.RM" "TRAIN53.RM" "TRAIN54.RM" "TRAIN55.RM" "TRAIN56.RM" "TRAIN57.RM" "TRAIN58.RM" "TRAIN59.RM"
[61] "TRAIN60.RM" "TRAIN61.RM" "TRAIN62.RM" "TRAIN63.RM" "TEST64.NB"  "TEST65.EW"  "TEST66.NA"  "TEST67.RM"  "TEST68.NA"  "TEST69.EW"
[71] "TEST70.BL"  "TEST71.NB"  "TEST72.NA"  "TEST73.RM"  "TEST74.NA"  "TEST75.EW"  "TEST76.NA"  "TEST77.NB"  "TEST78.BL"  "TEST79.NB"
[81] "TEST80.RM"  "TEST81.BL"  "TEST82.EW"  "TEST83.EW"  "TEST84.EW"  "TEST85.RM"  "TEST86.NB"  "TEST87.RM"  "TEST88.NB”

Data = Khan[,2:64]
Y = strsplit(names(Data),"\\.")
Y = unlist(Y)[ 2*(1:63)]
Y
 [1] "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "EW" "BL" "BL" "BL"
[27] "BL" "BL" "BL" "BL" "BL" "NB" "NB" "NB" "NB" "NB" "NB" "NB" "NB" "NB" "NB" "NB" "NB" "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM"
[53] "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM" "RM”

#  rF:  row F statistics and pvalues.  cF : conditional F statistics and pvalues.
pvalF= rF(Data,Y)
head(pvalF)
          F         pval LogFoldChange
1 15.046814 2.191233e-07     1.6731984
2 15.014638 2.250736e-07     0.5511725
3  7.232760 3.261875e-04     0.9340533
4  4.097402 1.040859e-02     0.6525674
5  1.884736 1.419950e-01     0.1345435
6   1.303855 2.816766e-01     1.0010550

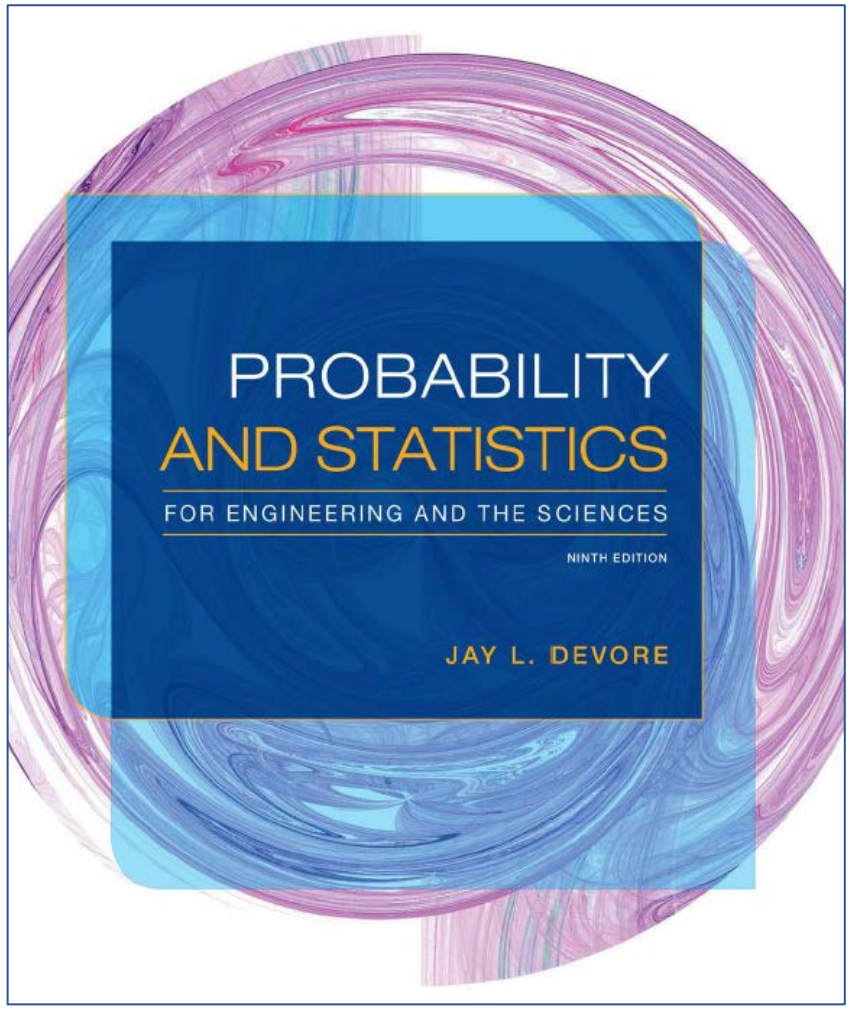


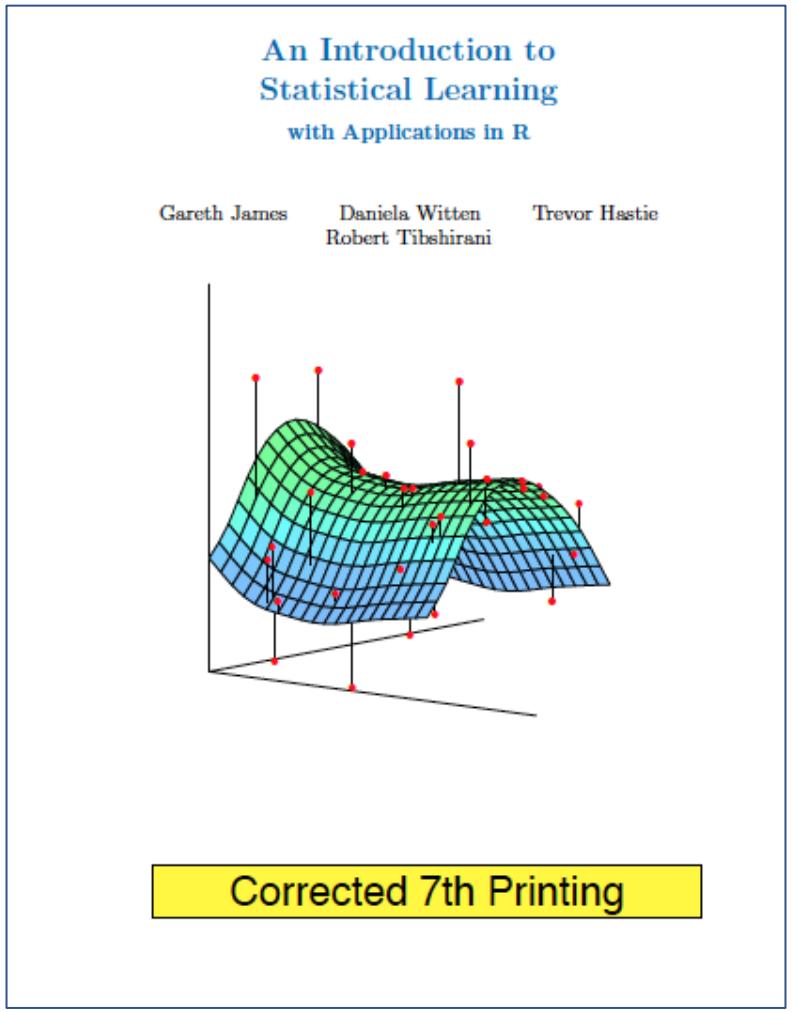
set.seed(25437)
pvalCF = cF(Data,Y))
plot(log(pvalCF),log(pvalF[,2]),pch=16,col=2)
abline(0,1,col=3,lwd=2)
```



Questions

References





https://www.statlearning.com/