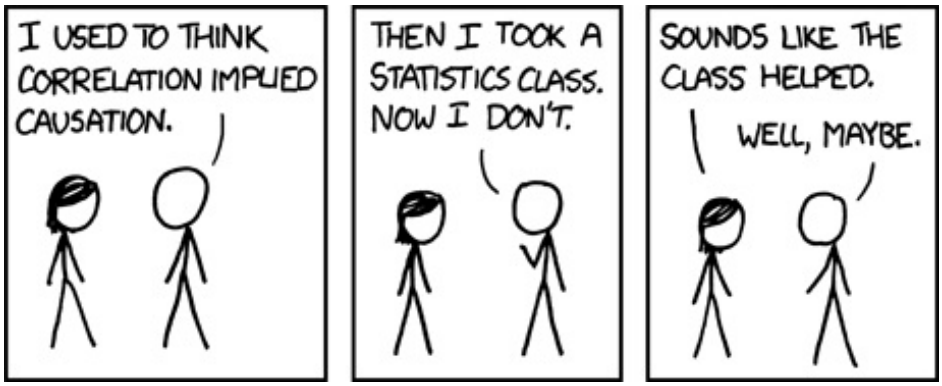
Correlation and Regression



http://xkcd.com/552/

Review

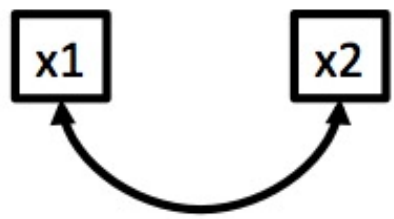
- Testing Hypotheses with P-Values
- Writing Functions
- Z, T, and χ^2 tests for hypothesis testing
- Power of different statistical tests using simulation

Today

1. Correlation
2. Mechanics of Simple Linear Regression
3. Basic Asssumptions of SLR

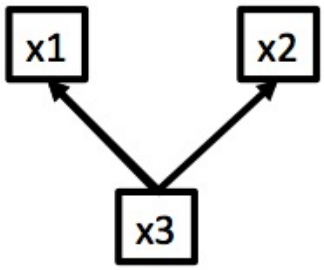
How are X and Y Related

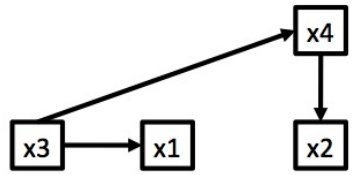
Causation (regression)
p(Y | X=x)

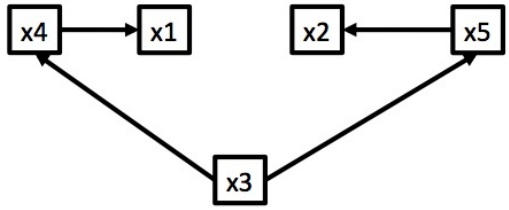


Correlation
p(Y=y, X=x)

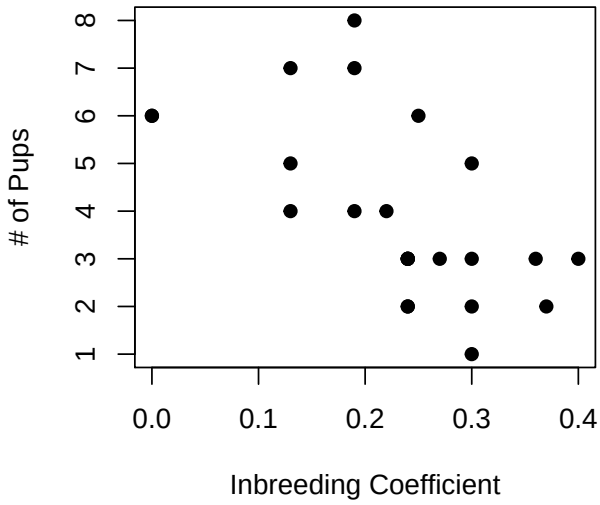
Correlation Can be Induced by Many Mechanisms



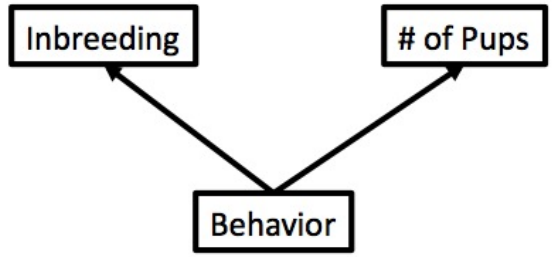




Example: Wolf Inbreeding and Litter Size



Example: Wolf Inbreeding and Litter Size



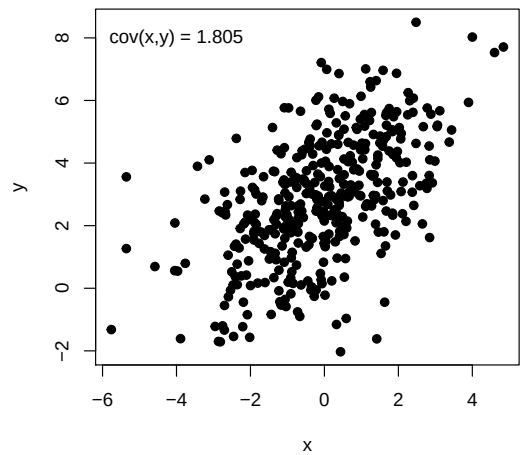
We don't know which is correct - or if another model is better. We can only examine *correlation*.

Covariance

Describes the relationship between two variables. Not scaled.

σ_{xy} = population level covariance, s_{xy} = covariance in your sample

$$\sigma_{XY} = \frac{\Sigma(X-\bar{X})(y-\bar{Y})}{n-1}$$

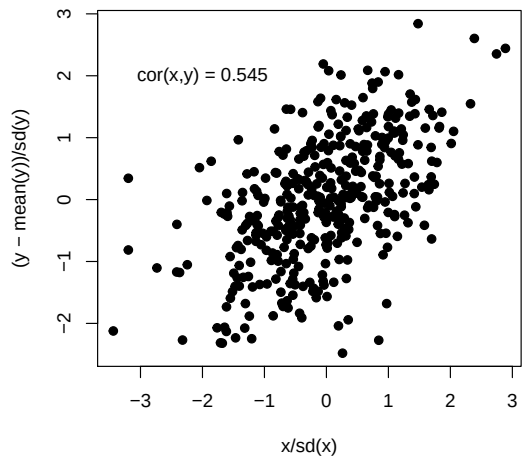


Pearson Correlation

Describes the relationship between two variables.
Scaled between -1 and 1.

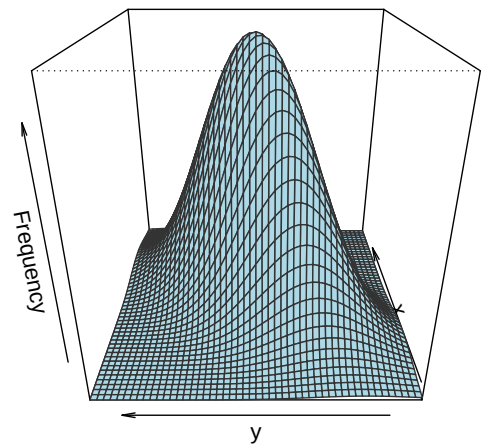
ρ_{xy} = population level correlation, r_{xy} = correlation in your sample

$$\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$$



Assumptions of Pearson Correlation

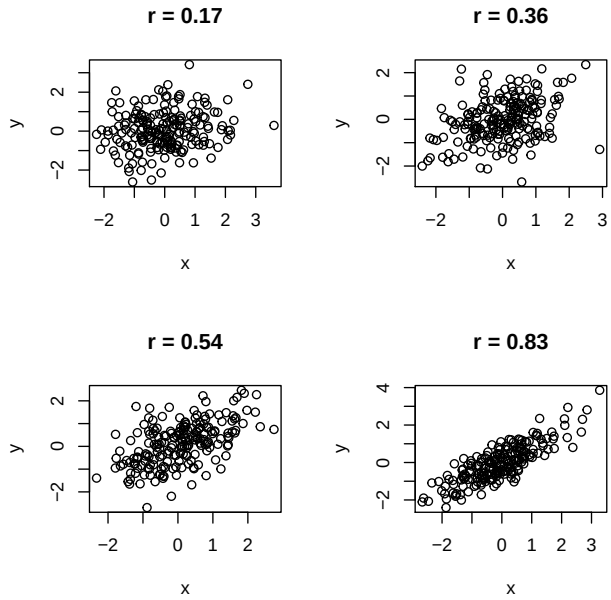
- Observations are from a **random sample**
- Each observation is **independent**
- X and Y are from a **Normal Distribution**



The meaning of r

Y is perfectly predicted by X if r = -1 or 1.

r^2 = the porportion of variation in y explained by x



Testing if r $\neq$ 0

Ho is r=0. Ha is r $\neq$ 0.

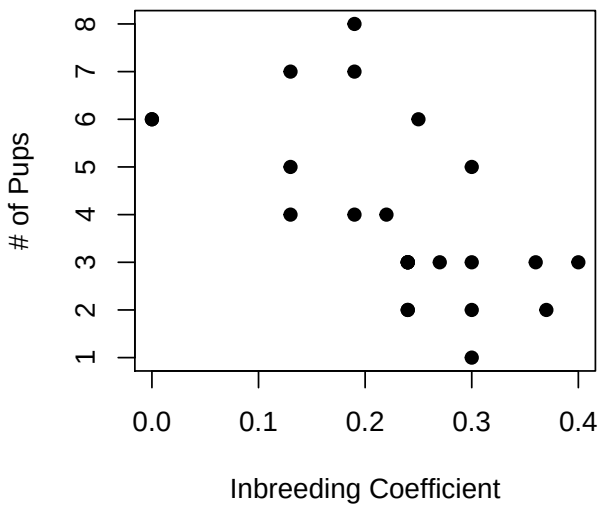
Testing: $t = \frac{r}{SE_r}$ **with df=n-2**

WHY n-2?

σ_{xy} Because you use two parameters: $\bar{X}$ and $\bar{Y}$

$$SE_r = \sqrt{\frac{1 - r^2}{n - 2}}$$

Example: Wolf Inbreeding and Litter Size



Example: Wolf Inbreeding and Litter Size

```
cov(wolves)

#                         inbreeding.coefficient    pups
# inbreeding.coefficient                0.009922 -0.1136
# pups                                 -0.113569  3.5199

cor(wolves)

#                         inbreeding.coefficient    pups
# inbreeding.coefficient                 1.0000 -0.6077
# pups                                  -0.6077  1.0000
```

Example: Wolf Inbreeding and Litter Size

```
with(wolves, cor.test(pups, inbreeding.coefficient))

#
#  Pearson's product-moment correlation
#
# data:  pups and inbreeding.coefficient
# t = -3.589, df = 22, p-value = 0.001633
# alternative hypothesis: true correlation is not equal to 0
# 95 percent confidence interval:
#  -0.8120 -0.2707
# sample estimates:
#     cor
# -0.6077
```

Exercise: Pufferfish Mimics & Predator Approaches

- Load up the pufferfish mimic data from W&S
- Plot the data
- Assess the correlation and covariance
- Assess Ho.
- Challenge - Evaluate Ha1: the correlation is 1.

Exercise: Pufferfish Mimics & Predator Approaches

```r
#get the correlation and se
puff_cor = cor(puffer)[1,2]
se_puff_cor = sqrt((1-puff_cor^2)/(nrow(puffer)-2))

puff_cor

# [1] 0.7767

t_puff0<-(puff_cor)/se_puff_cor

t_puff0

# [1] 5.232

2*pt(abs(t_puff0), nrow(puffer)-2, lower=FALSE)

# [1] 5.638e-05
```

Exercise: Pufferfish Mimics & Predator Approaches

```r
#get the correlation and se
cor.test(puffer$predators, puffer$resemblance)

#
#  Pearson's product-moment correlation
#
# data:  puffer$predators and puffer$resemblance
# t = 5.232, df = 18, p-value = 5.638e-05
# alternative hypothesis: true correlation is not equal to 0
# 95 percent confidence interval:
#  0.5092 0.9074
# sample estimates:
#    cor
# 0.7767
```

Exercise: Pufferfish Mimics & Predator Approaches

```
#t-test with difference from 1
t_puff<-(puff_cor-1)/se_puff_cor
t_puff

# [1] -1.504

#1 tailed, as > 1 is not possible
pt(t_puff, nrow(puffer)-2)

# [1] 0.07495
```

Violating Assumptions?

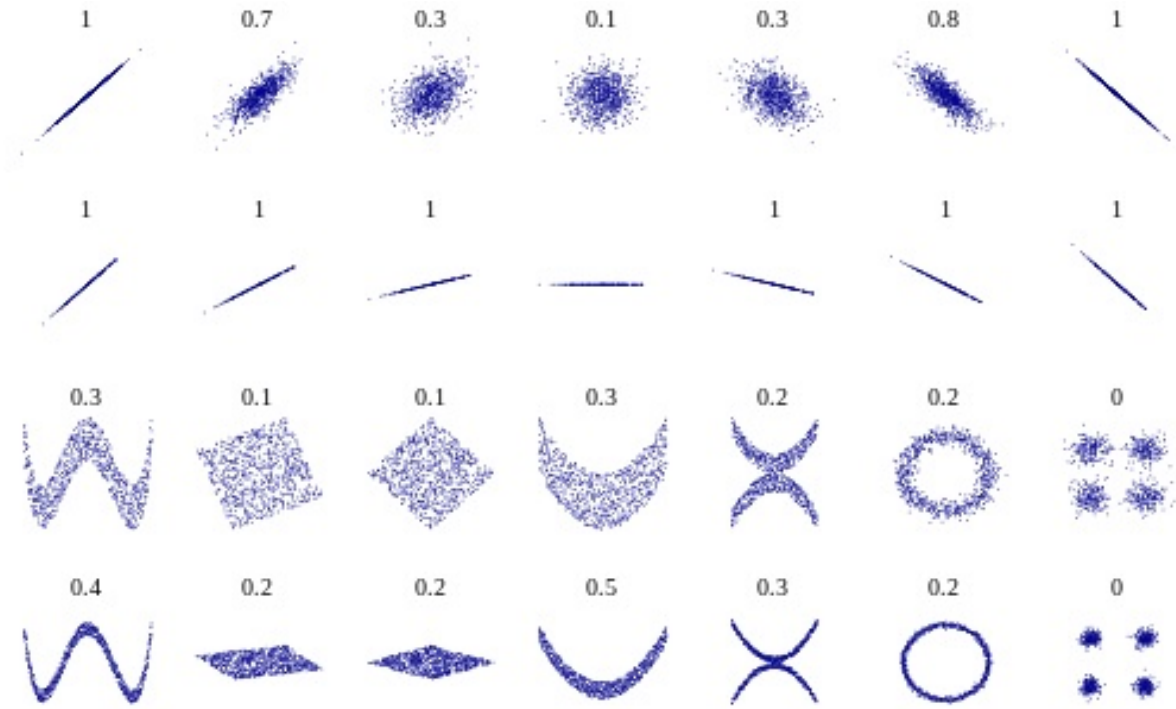
- Spearman's Correlation (rank based)
- Distance Based Correlation & Covariance (`dcor`)
- Maximum Information Coefficient (nonparametric)

All are lower in power for linear correlations

Spearman Correlation

1. Transform variables to ranks, i.e.,2,3... (`rank()`)
2. Compute correlation using ranks as data
3. If n $\leq$ 100, use Spearman Rank Correlation table
4. If n $>$ 100, use t-test as in Pearson correlation

Distance Based Correlation, MIC, etc.

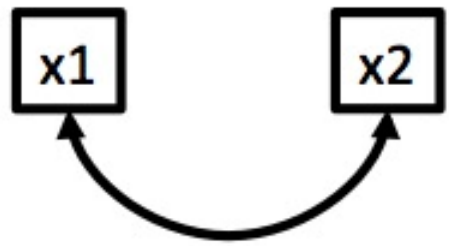


Least Squares Regression
$y = ax + b$

Then it's code in the data, give the keyboard a punch
Then cross-correlate and break for some lunch
Correlate, tabulate, process and screen
Program, printout, regress to the mean
-White Coller Holler by Nigel Russell

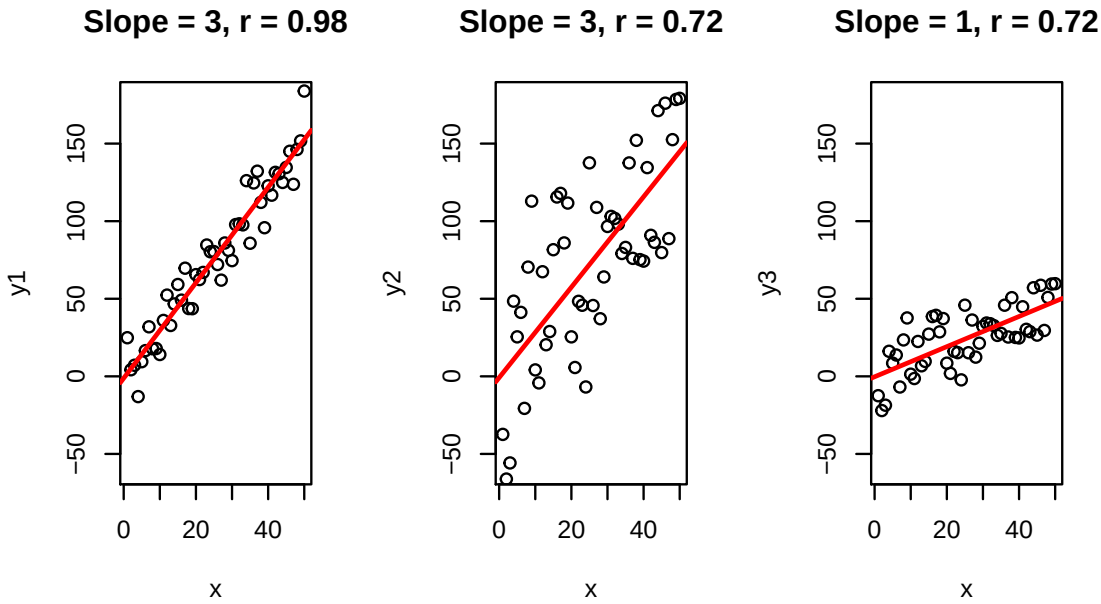
How are X and Y Related

Causation (regression)
$p(X2|X1 = x_1)$



Correlation
$p(X1 = x_1, X2 = x_2)$

Correlation v. Regression Coefficients

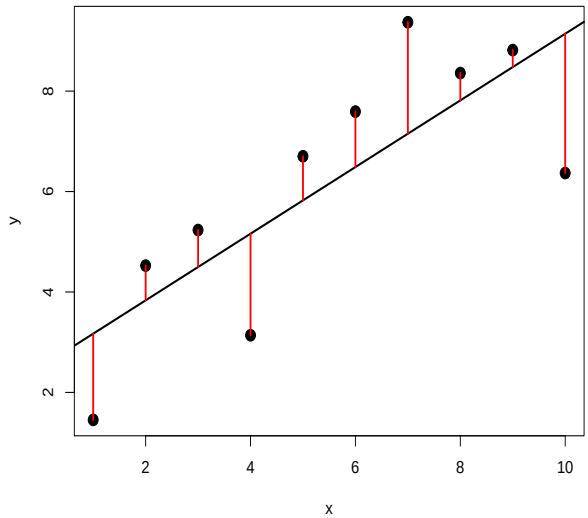


Basic Princples of Linear Regression

- Y is determined by X: p(Y | X=x)
- The relationship between X and Y is Linear
- The residuals of Y= a + bx are normally distributed (i.e., Y = a + bX + e where e $\sim$ N(0,σ))

Basic Principles of Least Squares Regression

$\widehat{Y} = a + bX$ where a = intercept, b = slope



Minimize Residuals defined as $SS_{residuals} = \sum(Y_i - \widehat{Y})^2$

Solving for Slope

$$b = \frac{s_{xy}}{s_x^2} = \frac{cov(x,y)}{var(x)}$$

$$= r_{xy}\frac{s_y}{s_x}$$

Solving for Intercept

Least squares regression line always goes through the mean of X and Y

$$\bar{Y} = a + b\bar{X}$$

$$a = \bar{Y} - b\bar{X}$$

Fitting a Linear Model in R

```
wolf_lm <- lm(pups ~ inbreeding.coefficient, data=wolves)
```

```
wolf_lm

#
# Call:
# lm(formula = pups ~ inbreeding.coefficient, data = wolves)
#
# Coefficients:
#             (Intercept)  inbreeding.coefficient
#                    6.57                  -11.45
```

Extracting Coefficients from a LM

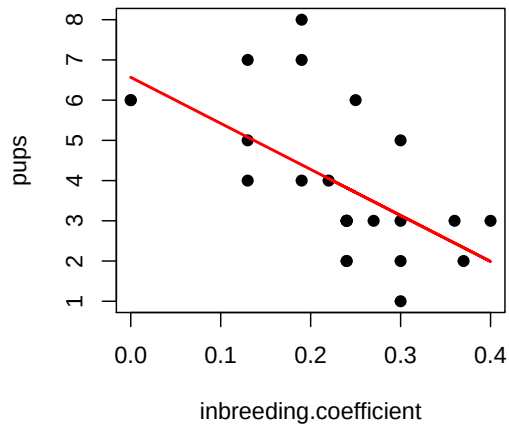
```
coef(wolf_lm)

#            (Intercept) inbreeding.coefficient
#                  6.567                -11.447

coef(wolf_lm)[1]

# (Intercept)
#       6.567
```

Extracting Fitted Values from a LM

```
fitted(wolf_lm)

#     1     2     3     4     5     6     7     8     9    10
# 6.567 6.567 5.079 5.079 5.079 4.392 4.392 4.392 3.706 3.820
#    11    12    13    14    15    16    17    18    19    20
# 3.820 3.820 3.820 3.820 3.820 3.477 3.133 3.133 3.133 3.133
#    21    22    23    24
# 2.446 1.989 2.332 4.049

coef(wolf_lm)[1] + coef(wolf_lm)[2]*0.25

# (Intercept)
#       3.706
```

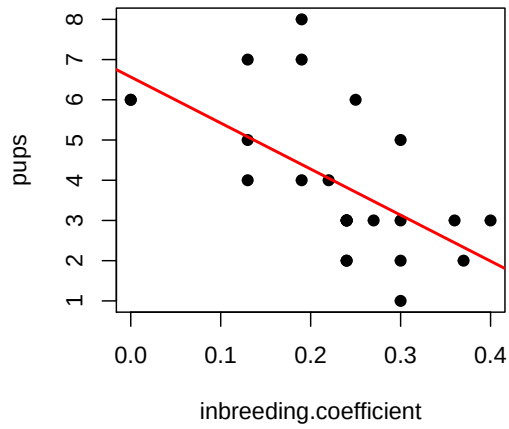
Plotting Fitted LMs

```
plot(pups ~ inbreeding.coefficient, data=wolves, pch=19)

matplot(wolves$inbreeding.coefficient, fitted(wolf_lm),
        add=T, lwd=2, col="red", type="l")
```



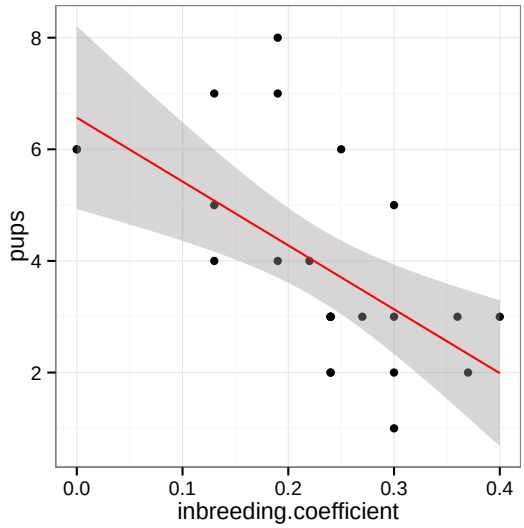
Plotting Fitted LMs

```
plot(pups ~ inbreeding.coefficient, data=wolves, pch=19)

abline(wolf_lm, col="red", lwd=2)
```



Ggplot2 and LMs

```
ggplot(data=wolves, aes(y=pups, x=inbreeding.coefficient)) +
  geom_point() +
  theme_bw() +
  stat_smooth(method="lm", color="red")
```



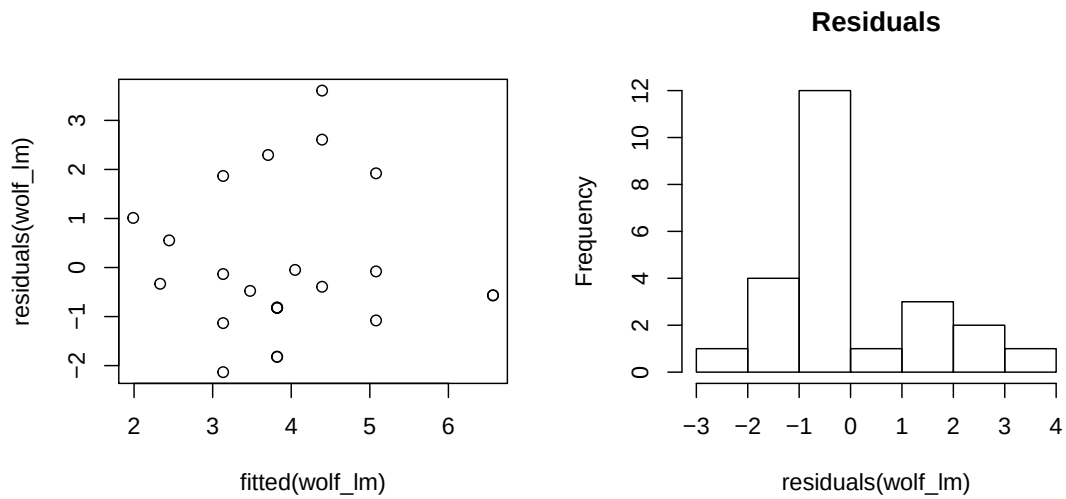
The Major Assumption of SLR

$$Y = a + bX + e$$

$$e \sim N(0,\sigma)$$

Checking Residuals

```
par(mfrow=c(1,2))
plot(fitted(wolf_lm), residuals(wolf_lm))
#
hist(residuals(wolf_lm), main="Residuals")
```

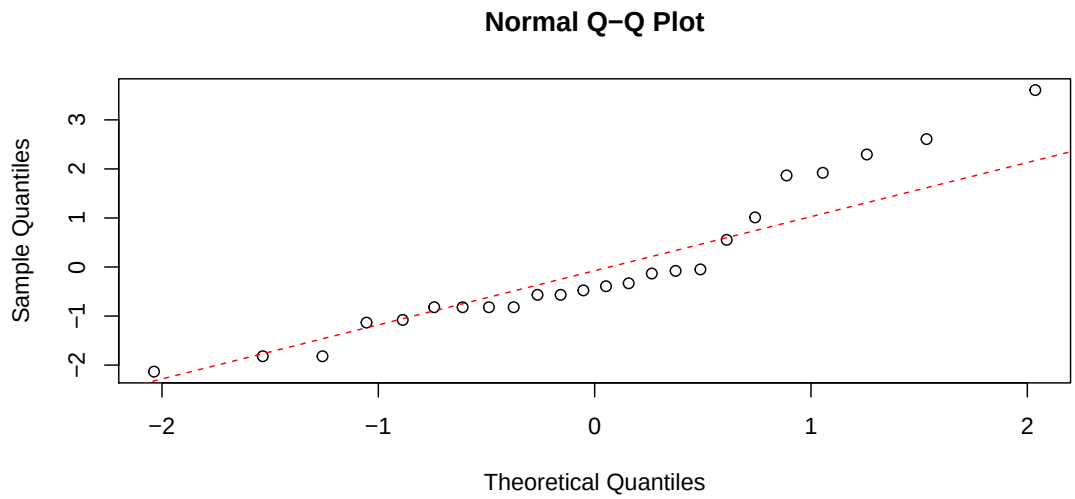


Checking for Normality of Residuals

```
#the quantiles in the data
probs <- 1:length(residuals(wolf_lm))/length(residuals(wolf_lm))

#now the data split up into those quantiles
quantile(residuals(wolf_lm), probs=probs)

#    4.167%    8.333%      12.5%    16.67%    20.83%       25%
# -1.83307  -1.82002  -1.21907  -1.08816  -0.87401  -0.82002
#    29.17%    33.33%     37.5%    41.67%    45.83%       50%
# -0.82002  -0.82002  -0.66202  -0.56722  -0.51815  -0.43449
#    54.17%    58.33%     62.5%    66.67%    70.83%       75%
# -0.36467  -0.24915  -0.11294  -0.06909   0.12679   0.66805
#    79.17%    83.33%     87.5%    91.67%    95.83%      100%
#   1.18964   1.87579   1.96755   2.32055   2.64931   3.60765
```

Checking for Normality of Residuals

```
qqnorm(residuals(wolf_lm))
qqline(residuals(wolf_lm), lty=2, col="red")
```



Exercise: Pufferfish Mimics & Predator Approaches

- Fit the pufferfish data
- Visualize the linear fit
- Examine whether there is any relationship between fitted values, residual values, and treatment