Putting your Regression Model to the Test

If it's possible to prove it wrong
You're going to want to know before too long
You'll need a test
- from *Put it to the Test* by They Might Be Giants

You have Fit a Model. Now...

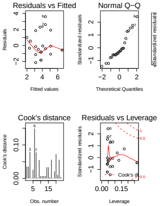
1. Can you really use this model fit?
2. Does your model explain variation in the data?
3. Are your coefficients different from 0?
4. How much variation is retained by the model?
5. How confident can you be in model predictions?

Assumptions of Ordinary Least Squares Regression

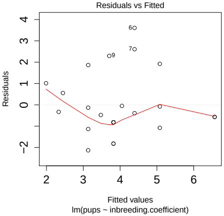
- Linearity
- Normality
- Results are not driven by outliers

Assumptions of Ordinary Least Squares Regression

```
par(mfrow = c(2, 3))
plot(wolf_lm, which = 1:5, cex.axis = 1.4)
```

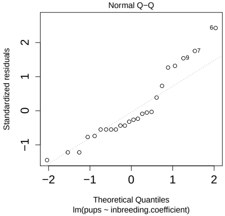


Is Anything Systematically Wrong?



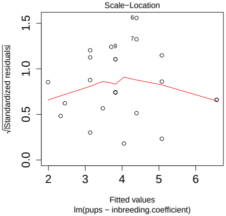
- Patterns produced if relationship isn't linear
- Other drivers may affect high or low values

Are the Residuals Normal?



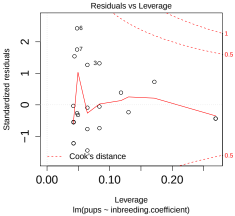
- Quantile-Quantile (QQ) Plot
- If residuals are normal, should fall on line

Standardized Residuals to Diagnose Error Distribution Problems



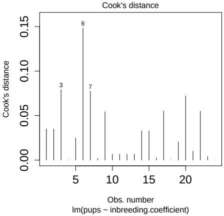
- Residuals are standardized
- Shape to data indicates deviation from normality
- Wedge shapes, bow-ties, trends all indicated problems

Influential Observations



- Leverage is distance from mean X
- $h = \frac{1}{n} + \frac{(x_i - \bar{x})^2}{SS_x}$

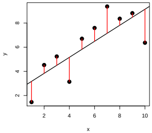
Influential Observations



- Combines leverage and residual properties
- Larger values, greater effect on results

Testing the Model

Ho = The model predicts no variation in the data.
Ha = The model predicts variation in the data.



$$SS_{Total} = SS_{Regression} + SS_{Error}$$

Components of the Total Sums of Squares

$$SS_R = \sum(\hat{Y}_i - \bar{Y})^2, \text{ df=1}$$

$$SS_E = \sum(Y_i - \hat{Y}_i)^2, \text{ df=n-2}$$

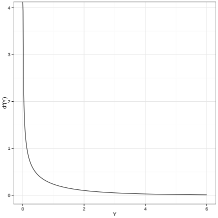
To compare them, we need to correct for different DF. This is the Mean Square.

MS=SS/DF

e.g, $MS_E = \frac{SS_E}{n-2}$

F Test to Evaluate Predictor's Contribution

$F = \frac{MS_R}{MS_E}$ with DF=1,n-2



1-Tailed Test

F-test Example: Wolves

```
anova(wolf_lm)

## Analysis of Variance Table
##
## Response: pups
##                       Df Sum Sq Mean Sq F value Pr(>F)
## inbreeding.coefficient  1   29.9   29.90    12.9 0.0016
## Residuals              22   51.1    2.32
```

Error in the Slope Estimate

$$SE_b = \sqrt{\frac{MS_E}{SS_X}}$$

$$95\%\ \text{CI} = b \pm t_{\alpha(2),df} SE_b$$

Assessing the Slope

$$t_b = \frac{b - \beta_0}{SE_b}$$

DF=n-2

Coefficient of Determination

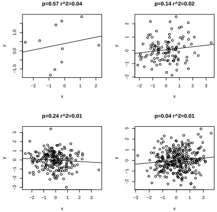
R^2 = The porportion of Y is predicted by X.

$$R^2 = \frac{SS_{regression}}{SS_{total}}$$

$$= 1 - \frac{SS_{regression}}{SS_{error}}$$

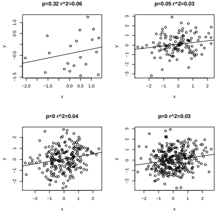
The "Obese N"

High sample size can lead to a low p-value, even if no association exists



Sample Size and R^2

High sample size can lead to a low R^2 if residual SD is high relative to slope



Example: Wolf Pups

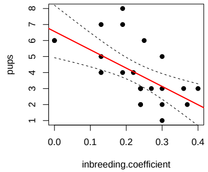
```
summary(wolf_lm)

##
## Call:
## lm(formula = pups ~ inbreeding.coefficient, data = wolves)
##
## Residuals:
##     Min      1Q Median     3Q    Max
## -2.133 -0.820 -0.434  0.668  3.608
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.567      0.791    8.31  3.1e-08
## inbreeding.coefficient  -11.447      3.189   -3.59   0.0016
##
## Residual standard error: 1.52 on 22 degrees of freedom
## Multiple R-squared: 0.369, Adjusted R-squared: 0.341
## F-statistic: 12.9 on 1 and 22 DF,  p-value: 0.00163
```

Exercise: Pufferfish Mimics & Predator Approaches

- Fit the pufferfish data
- Evaluate whether it meets assumptions
- Evaluate Ho and how well this model explains the data

Confidence Intervals Around Fit

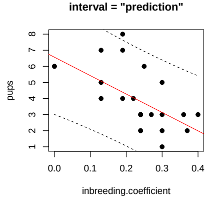


Accomodates uncertainty in slope & intercept

Confidence Intervals Around Fit

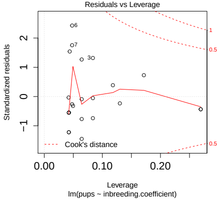
```
plot(pups ~ inbreeding.coefficient, data=wolves, pch=19)
abline(wolf_lm, col="red", lwd=2)
predFrame <- data.frame(inbreeding.coefficient=seq(0,0.4,.01))
predFitConf <-  predict(wolf_lm, newdata=predFrame,
                        interval="confidence")
matlines(predFrame, predFitConf[,2:3], type="l", lty=2, col="black")
```

Confidence Intervals Around Prediction



Remember: Extrapolation beyond range of data is bad practice

Testing the Effect of Removing Outliers



Testing the Effect of Removing Outliers

```
wolf_lm_sub <- lm(pups ~ inbreeding.coefficient,
                  data=wolves, subset=-c(6,7,3))

#another way
wolf_lm_sub <- update(wolf_lm, subset=-c(6,7,3))
```

Comparing Two Slopes

Ho: $\beta_1 = \beta 2$

$$t = \frac{(b_1 - b_2) - (\beta_1 - \beta 2)}{SE_{b1-b2}}$$

df $=$ n1 - 2 $+$ n2 - 2

Comparing Two Slopes

$$SE_{b1-b2} = \sqrt{\frac{MSE_p}{SS_{X1}} + \frac{MSE_p}{SS_{X2}}}$$

$$MSE_p = \frac{SSE_1 + SSE_2}{DF}$$

Comparing Two Slopes

```
wolf_lm_sub <- lm(pups ~ inbreeding.coefficient, data = wolves,
    subset = -c(6, 7, 3))

# another way
wolf_lm_sub <- update(wolf_lm, subset = -c(6, 7, 3))
```

Comparing Two Slopes

```
#get anova tables for later extraction of MSE
a1 <- anova(wolf_lm)
a2 <- anova(wolf_lm_sub)

#We'll need Sums of Squres from each set of X's
with(wolves, {
  ss1 <<- sum((inbreeding.coefficient -
              mean(inbreeding.coefficient))^2)

  ss2 <<- sum((inbreeding.coefficient[-c(6,7,3)] -
              mean(inbreeding.coefficient[-c(6,7,3)]))^2)
})
```

Comparing Two Slopes

```
# calculate the DF
df <- nrow(wolves) * 2 - 3 - 4

# calcaulate the mean square pooled error
msp <- (a1[2, 3] + a2[2, 3])/(df)

# calculate the SE of the difference
sep <- sqrt(msp/ss1 + msp/ss2)

# calculate t
t <- (coef(wolf_lm)[2] - coef(wolf_lm_sub)[2])/sep

# get the p value
pt(t, df) * 2

## inbreeding.coefficient
##                0.01322
```

Exercise: Pufferfish Mimics & Predator Approaches

- Check confidence and prediction intervals of the puffer fit
- Evaluate the effect of dropping outliers
- Challenge: write a function to compare slopes from two different lms