

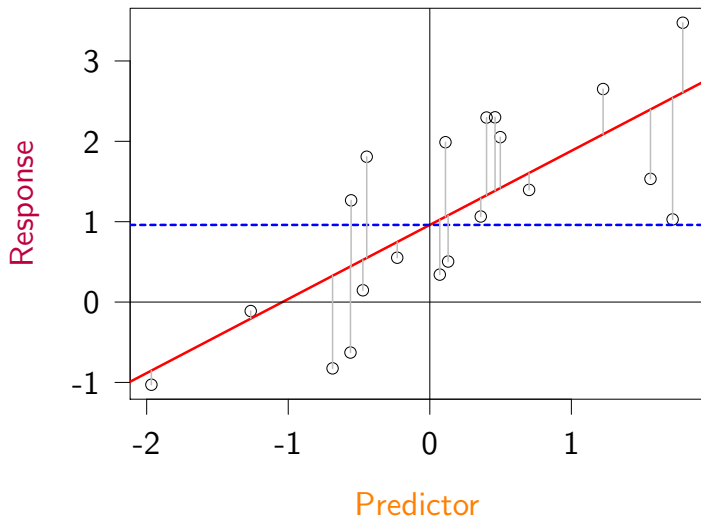
What on earth is going on with my linear models??!

March 8, 2018

- 1 Linear model, reminder
- 2 Diagnostics
- 3 A puzzling but simple problem: Over-fit and collinearity
- 4 Heteroschedasticity: the spooky word
- 5 Cures

# A simple linear model

$$\text{Response} = \text{Intercept} + \text{Slope} \times \text{Predictor} + \text{Error}$$



# A simple linear model

$$\text{Response} = \text{Intercept} + \text{Slope} \times \text{Predictor} + \text{Error}$$

In R:

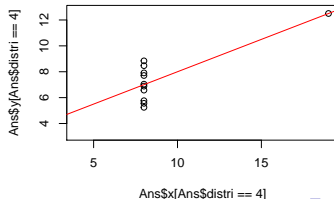
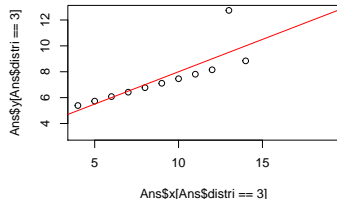
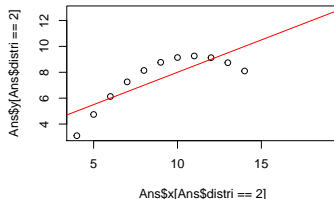
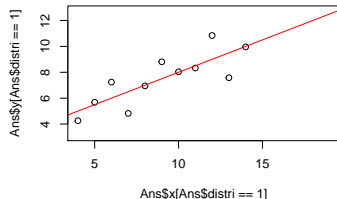
```
lm(response ~ 1 + predictor1 + predictor2, data=data)
# equivalent to
lm(response ~ predictor1 + predictor2, data=data)
```

- Intercept can be explicit or implicit
- Can remove intercept with  $\dots \sim 0 + \dots$
- Error is implicit
- Feed the option `data=` to keep code short, reliable and flexible
- Order of predictors do not matter

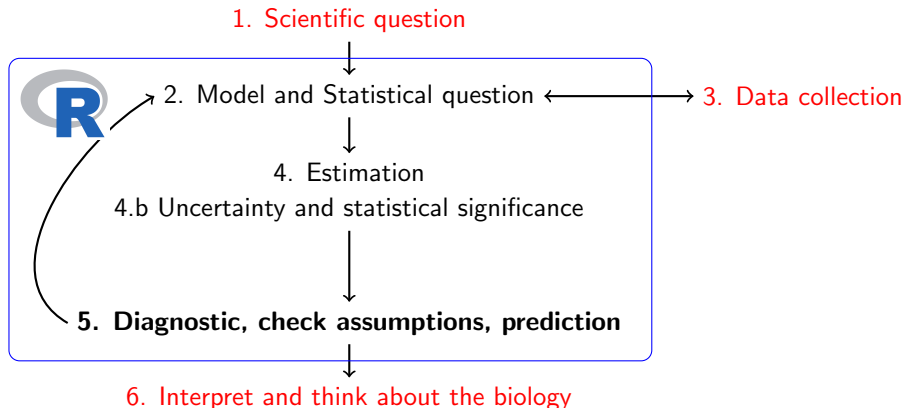
- 1 Linear model, reminder
- 2 **Diagnostics**
- 3 A puzzling but simple problem: Over-fit and collinearity
- 4 Heteroschedasticity: the spooky word
- 5 Cures

# Why we need checks: summary(lm) isn't enough

```
Ans <- read.csv(file = "Anscombe.csv")
```



# General approach



# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless; e.g. static allometry*



# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless; e.g. static allometry*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless; e.g. static allometry*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless; e.g. static allometry*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution  
*Risk: Poor predictions*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless; e.g. static allometry*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution  
*Risk: Poor predictions*
- Homoscedasticity (constant error variance)  
*Risk: Over-optimistic uncertainty, unreliable predictions*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless; e.g. static allometry*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution  
*Risk: Poor predictions*
- Homoscedasticity (constant error variance)  
*Risk: Over-optimistic uncertainty, unreliable predictions*
- Independence of error  
*Risk: Bias and over-optimistic uncertainty*

# Why we need checks: missing a relationship

```
forprediction <- read.csv(file = "forprediction.csv")
```

Does "predictor" predict "obs"?

# Why we need checks: missing a relationship

```
forprediction <- read.csv(file = "forprediction.csv")
```

Does "predictor" predict "obs"?

```
summary(lm(obs ~ 1 + predictor, data=forprediction) )
```

# Why we need checks: missing a relationship

Does "predictor" predict "obs"? Apparently not:

```
summary(lm(obs ~ 1 + predictor, data=forprediction) )
```

Call:

```
lm(formula = obs ~ 1 + predictor, data = forprediction)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.1962	-0.5326	0.1378	0.5785	1.8664

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.72530	0.16953	16.076	<2e-16 ***
predictor	-0.01129	0.02956	-0.382	0.703

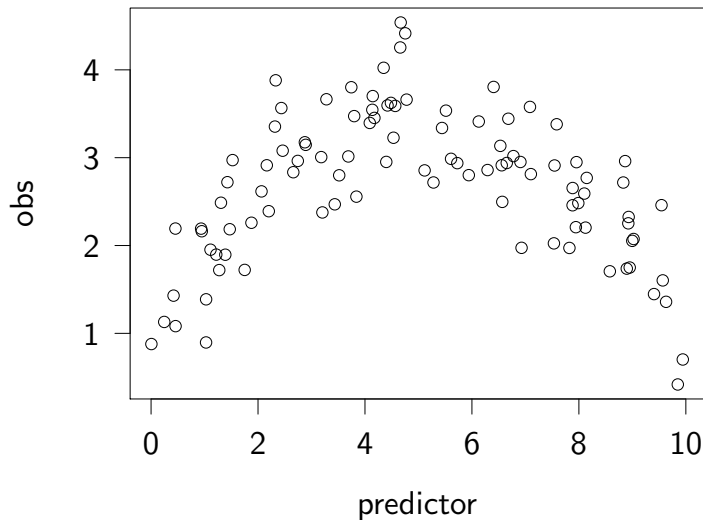
---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8382 on 98 degrees of freedom



# Why we need checks: missing a relationship



# How to check?

```
m0 <- lm(obs ~ 1 + predictor, data=forprediction)
summary(m0)
```

Call:

```
lm(formula = obs ~ 1 + predictor, data = forprediction)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-2.1962	-0.5326	0.1378	0.5785	1.8664

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	2.72530	0.16953	16.076	<2e-16 ***
predictor	-0.01129	0.02956	-0.382	0.703

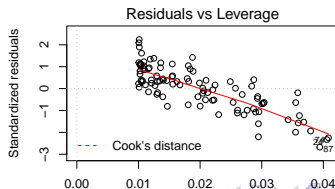
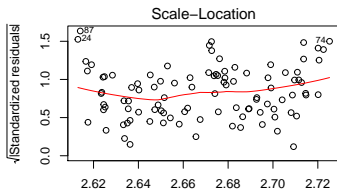
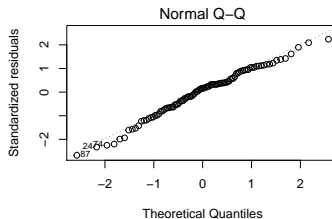
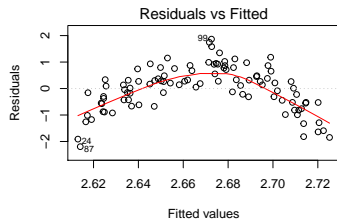
---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8382 on 98 degrees of freedom

# How to check?

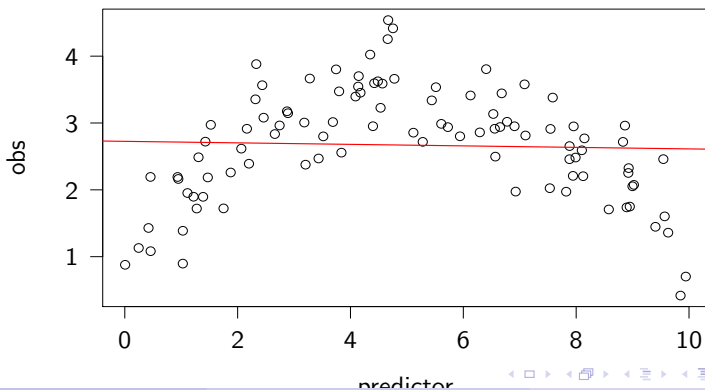
```
par(mfrow=c(2,2))  
plot(m0)
```



# How to check?

```
m0 <- lm(obs ~ 1 + predictor, data=forprediction)
```

```
setPar()  
plot(x=forprediction$predictor, y=forprediction$obs, xlab="predictor",  
     abline(m0, col="red", lwd=3) #simple prediction, without SE
```



# Check checklist

- **Visualize your data**
- Residual in summary(): are they symmetrical?
- plot(lm):
  - ① trend residual/fitted?
  - ② Normal residuals?
  - ③ trend in residual variance?
  - ④ outliers?
- Predictions: range and biological meaning

# Fix?

```
lm(obs ~ 1 + predictor , data=forprediction)
```

# Fix?

```
lm(obs ~ 1 + predictor , data=forprediction)
```

```
m1 <- lm(obs ~ 1 + predictor + I(predictor^2), data=forprediction)  
plot(m1)
```

How about prediction? (abline(m1) won't work here)

# Introduction to prediction

```
m1 <- lm(obs ~ 1 + predictor + I(predictor^2), data=forprediction)
coef(m1)
```

(Intercept)	predictor	I(predictor^2)
1.07782598	0.94760312	-0.09524182

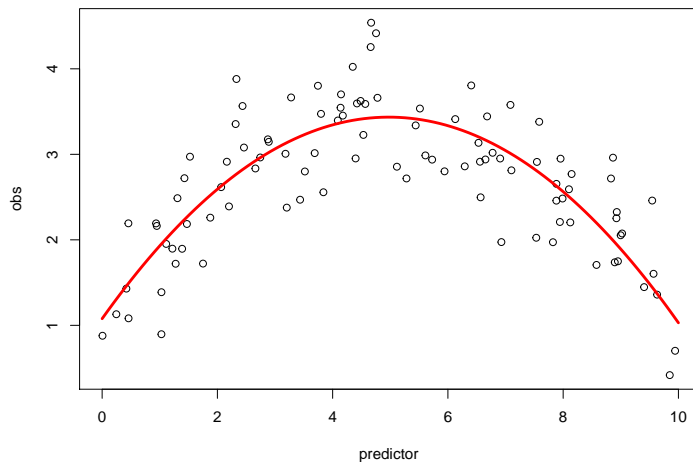
## Small exercise

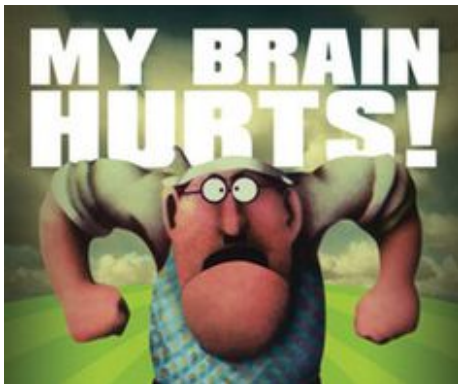
Use regression coefficients to predict "obs" from "predictor", and add a prediction line on the plot obs/predictor.

Is the fit satisfactory?



# Introduction to prediction





- 1 Linear model, reminder
- 2 Diagnostics
- 3 A puzzling but simple problem: Over-fit and collinearity**
- 4 Heteroschedasticity: the spooky word
- 5 Cures

# Over-fit and collinearity

## Small exercise

Load Cdata.csv, fit models of  $y$  predicted by  $x_1$  and  $x_2$ , or  $x_2$  and  $x_3$ .  
Something is weird, what is going on? What to do?

- 1 Linear model, reminder
- 2 Diagnostics
- 3 A puzzling but simple problem: Over-fit and collinearity
- 4 Heteroschedasticity: the spooky word**
- 5 Cures

# Exponential data

```
plot(x, obs, ylim = c(-20, max(obs)))  
abline(lm(obs ~ x))  
summary(lm(obs ~ x))  
plot(lm(obs ~ x))
```

# Exponential data: prediction

Make a prediction over the range of  $x$ , with prediction interval

# Exponential data: prediction

Make a prediction over the range of  $x$ , with prediction interval

```
Xnewdata <- data.frame(x=seq(from=min(x), to=max(x),  
                             length.out = 100))  
Xpred <- predict(object = lm(obs ~ x), newdata = Xnewdata,  
                 se.fit = TRUE, interval = "prediction")
```



# Exponential data: prediction

Make a prediction over the range of  $x$ , with prediction interval

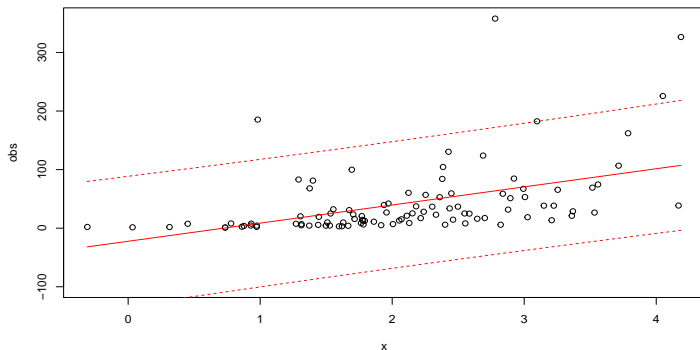
```
Xnewdata <- data.frame(x=seq(from=min(x), to=max(x),  
                             length.out = 100))  
Xpred <- predict(object = lm(obs ~ x), newdata = Xnewdata,  
                 se.fit = TRUE, interval = "prediction")
```

```
Xnewdata <- cbind(Xnewdata, Xpred)
```

```
head(Xnewdata)
```

# Exponential data: prediction

```
plot(x, obs, ylim = c(-100, max(obs)))  
lines(Xnewdata$x, Xnewdata$fit.fit, col="red")  
lines(Xnewdata$x, Xnewdata$fit.lwr, col="red", lty=2)  
lines(Xnewdata$x, Xnewdata$fit.upr, col="red", lty=2)
```



# Exponential data: confidence

**Prediction interval:** Where the model predicts new data would be sampled, including variation unrelated to predictor

# Exponential data: confidence

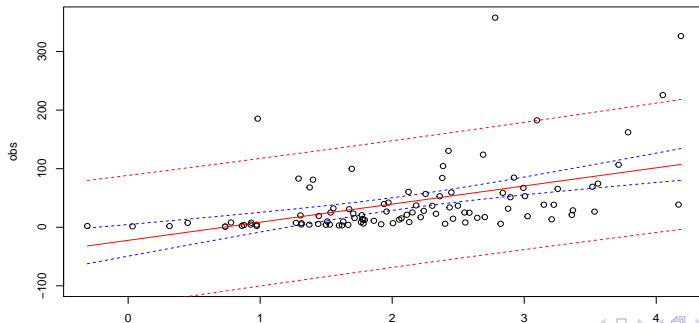
**Prediction interval:** Where the model predicts new data would be sampled, including variation unrelated to predictor

**Confidence interval:** Where your model predicts the MEAN new data would be sampled; or where is the true relationship with predictor

```
Xconf <- predict(object = lm(obs ~ x), newdata = Xnewdata,  
                 se.fit = TRUE, interval = "confidence")  
Xnewdata[,c("conf.lwr", "conf.upr")] <- Xconf$fit[,2:3]
```

# Exponential data: confidence

```
plot(x, obs, ylim = c(-100, max(obs)))  
lines(Xnewdata$x, Xnewdata$fit.fit, col="red")  
lines(Xnewdata$x, Xnewdata$fit.lwr, col="red", lty=2)  
lines(Xnewdata$x, Xnewdata$fit.upr, col="red", lty=2)  
lines(Xnewdata$x, Xnewdata$conf.lwr, col="blue", lty=2)  
lines(Xnewdata$x, Xnewdata$conf.upr, col="blue", lty=2)
```



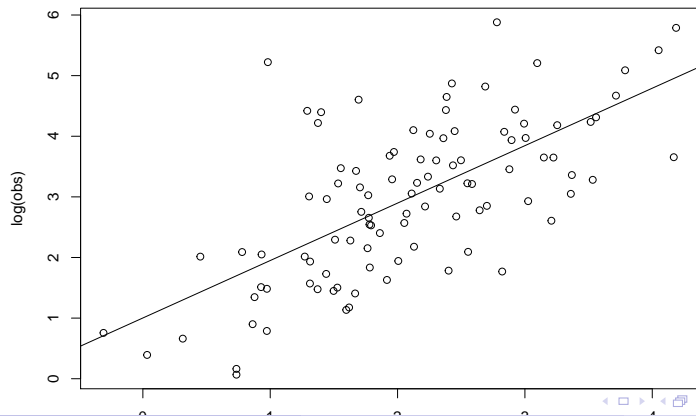
# Exponential data: fix

What would you suggest?

# Exponential data: fix

What would you suggest? Log-transform

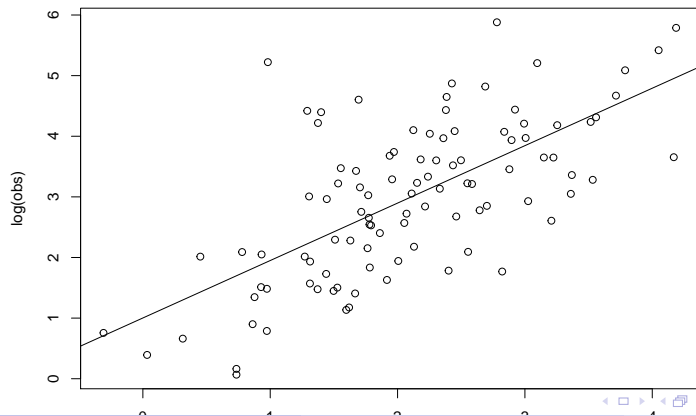
```
plot(x, log(obs))  
abline(lm(log(obs) ~ x))
```



# Exponential data: fix

What would you suggest? Log-transform

```
plot(x, log(obs))  
abline(lm(log(obs) ~ x))
```





# Practice lm() with parasites

## What explains variation in parasitic load?

You collected ecto-parasites on some furry large mammals at three locations. Parasites break easily when we collect them and are impossible to count, so we decide to measure parasitic load as their mass. **Why do some mammals have larger parasitic load?**

# Practice `lm()` with parasites

## What explains variation in parasitic load?

You collected ecto-parasites on some furry large mammals at three locations. Parasites break easily when we collect them and are impossible to count, so we decide to measure parasitic load as their mass. **Why do some mammals have larger parasitic load?**

- Load the `Para.csv` data (don't forget: `str()`, `summary()`, `plot()`...)
- Model `Parasite_Mass` using `lm()`
- Find what variables predict `Parasite_Mass`
- How good are your models? Assumptions? Prediction?
- What biological interpretation can you imagine?

- 1 Linear model, reminder
- 2 Diagnostics
- 3 A puzzling but simple problem: Over-fit and collinearity
- 4 Heteroschedasticity: the spooky word
- 5 Cures

# Multiple regression

# Transformations

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*



# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution  
*Risk: Poor predictions*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution  
*Risk: Poor predictions*
- Homoscedasticity (constant error variance)  
*Risk: Over-optimistic uncertainty, unreliable predictions*

# Linear model basic assumptions

Not necessarily wrong, but typical interpretation assumes:

- Linear combination of parameters (including transformation, polynoms, interactions. . . )  
*Risk: biologically meaningless*
- Predictor not perfectly correlated  
*Risk: Model won't run, unstable convergence, or huge SE*
- Measurement error in predictors  
*Risk: bias estimates (underestimate with Gaussian error)*
- Gaussian error distribution  
*Risk: Poor predictions*
- Homoscedasticity (constant error variance)  
*Risk: Over-optimistic uncertainty, unreliable predictions*
- Independence of error  
*Risk: Bias and over-optimistic uncertainty*

# What do you want to learn about?

## Topics

