

Stesh went to Hawaii!



This is a picture taken on my vacation in 2017 to Hawaii. This picture was taken on the main island of Oahu at Kualoa Ranch. This valley has been filmed in many movies including Jurassic Park, Jurassic World, Jumanji, King Kong, and Godzilla to name a few.

Linear models

Data Science for Biologists

Dr. Spielman

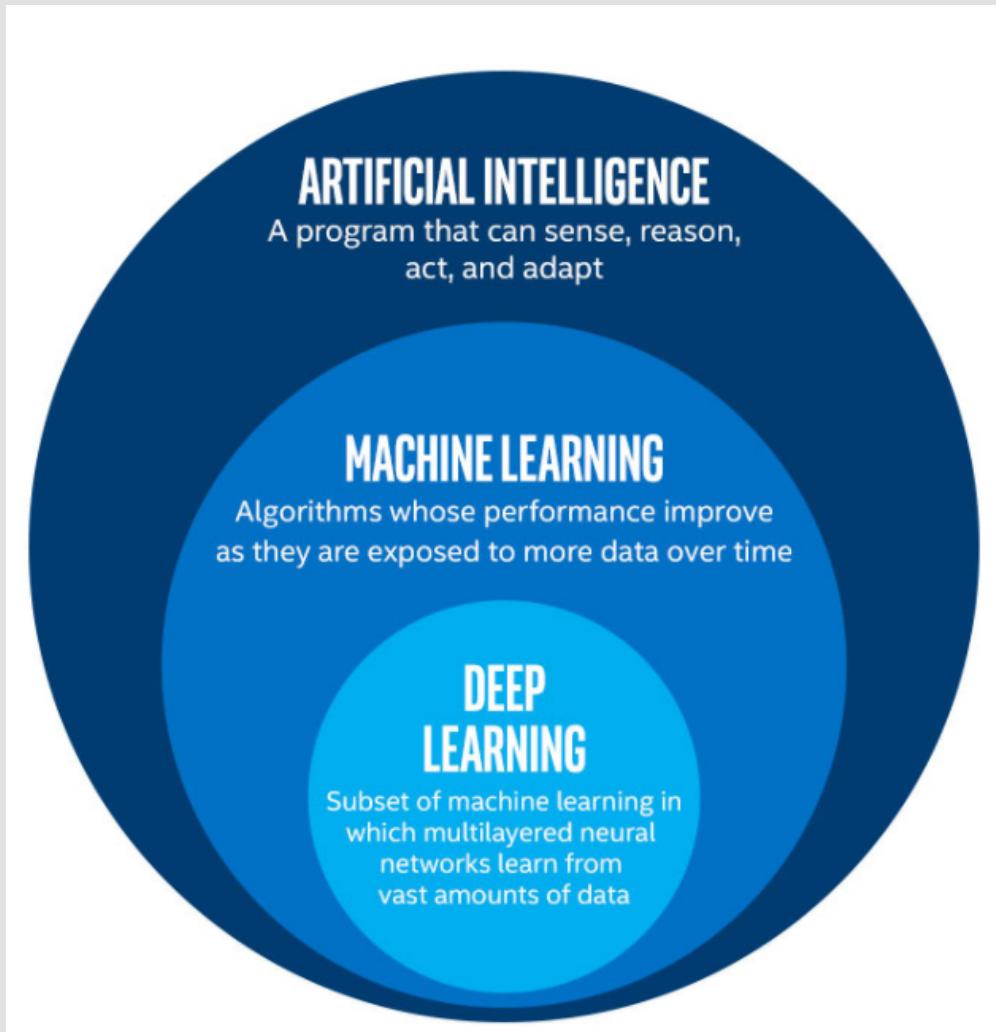
CAVEAT: This is not a statistics class

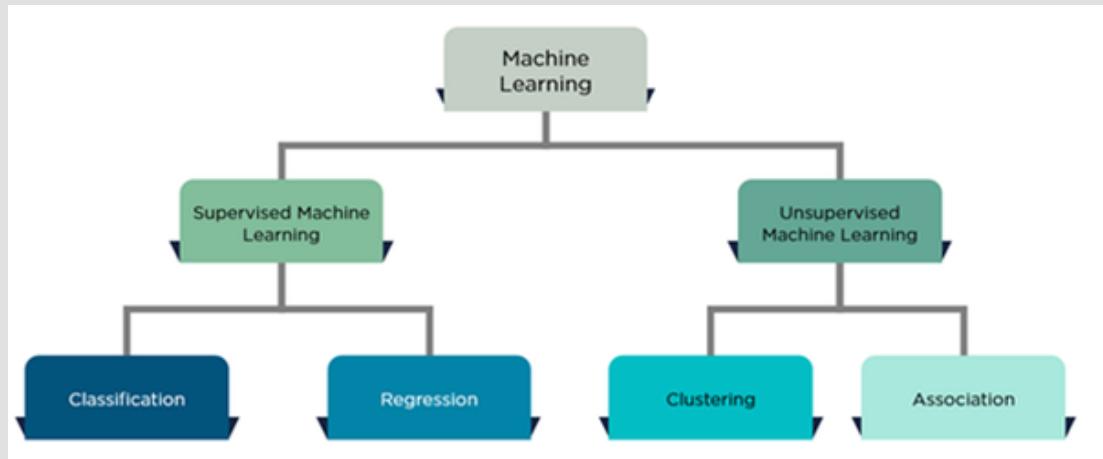
We will learn *to perform analyses and interpret results* using a few common modeling approaches used in data science industry

We will NOT be diving into the technical derivations or gnarly innards of the statistics of these models

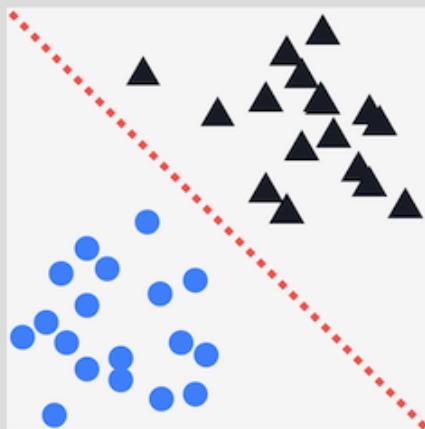
If you want to pursue data science down the line, *you will eventually need to know the technical aspects too.*

Machine Learning and AI

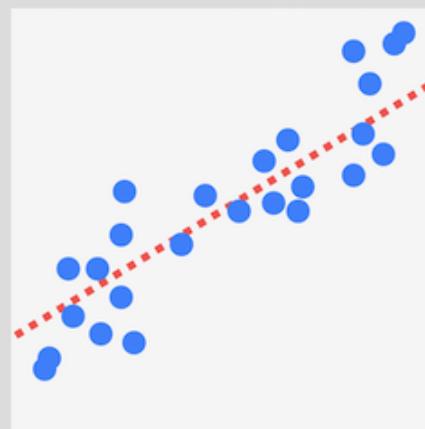




Classification



Regression



Everything you've learned is a ~~drum path~~ linear model!

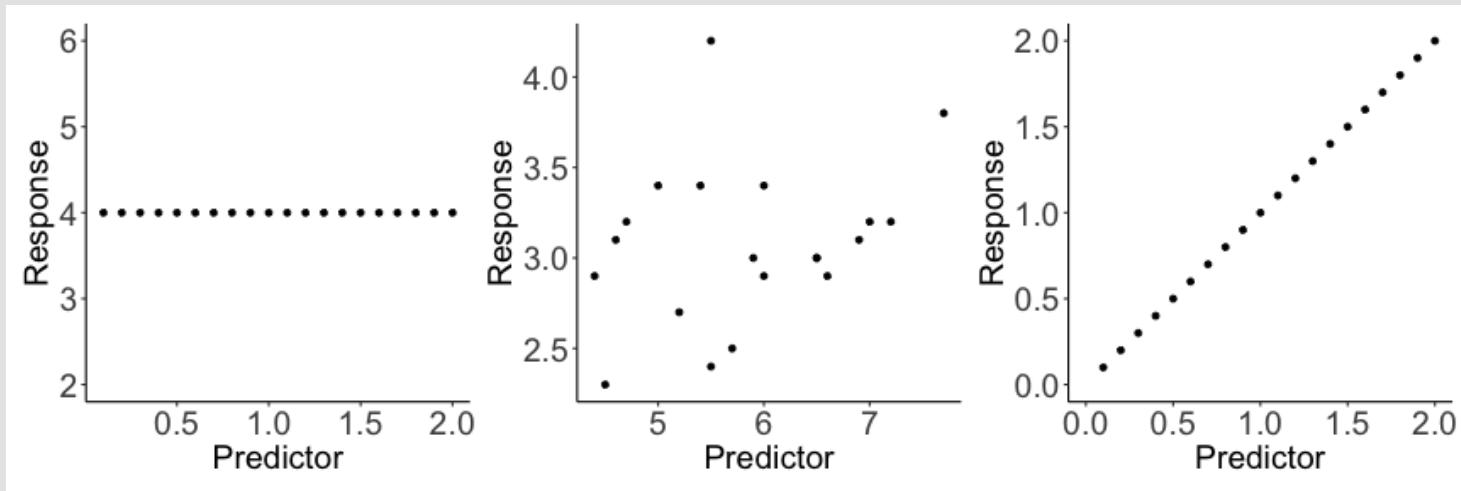
- Correlation
 - Pearson's correlation is what you know: $-1 < r < 1$
- Regression
- ANOVA, ANCOVA, MANOVA, etc.
- t -tests and X^2 -tests
- Wilcoxon/Mann-Whitney U tests, sign tests

Want to prove this to yourself? (advanced): <https://lindeloev.github.io/tests-as-linear/>

The goal of linear modeling is to *explain variation in a variable of interest*

In the applied "machine learning" world, we also use models to *predict future outcomes for that variable of interest*

What does "explain variation" mean?



We will learn two types of *generalized linear models*

Linear regression/model: Use this method when the response is a **numeric variable**

- Key assumptions:
 - Any numeric predictors are linearly related to the response
 - The values of the response variable have equal variance across categories of any categorical predictor
 - The *residuals* of the model are normally distributed. There is NO REQUIREMENT for the data itself to follow a normal distribution
 - Predictors should be independent of one another

Logistic regression/model: Use this method when the response is a **binary variable**

- We'll learn this next!

What is the goal of GLMs?

- We want to explain variation in a *response variable* using all suitable predictors
 - How does one determine "suitable predictors"?

Hypothesis-testing ("science")

- Predictors are based on experimental setup
- *Specific* goal of knowing how those *specific* predictors affect response
- Not necessarily interested in seeing how model performs on new data it has never seen

Exploratory ("industry"/"big-data")

- You have *a bunch* of data and need to figure out, which predictors should I use in my model to best explain the response?
- Less likely you care about specific effects of individual predictors
- More likely to be interested in applying the model to new data for prediction

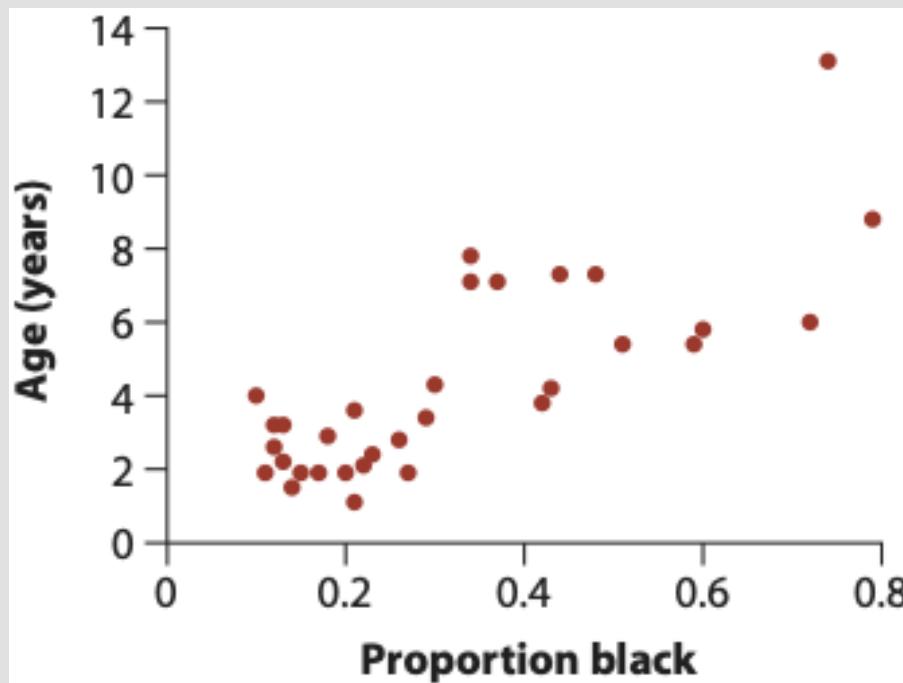
Linear models

$$Y = \beta_1 X_1 + \beta_0 + \epsilon$$

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \dots + \beta_N X_N + \epsilon$$

Simple linear regression

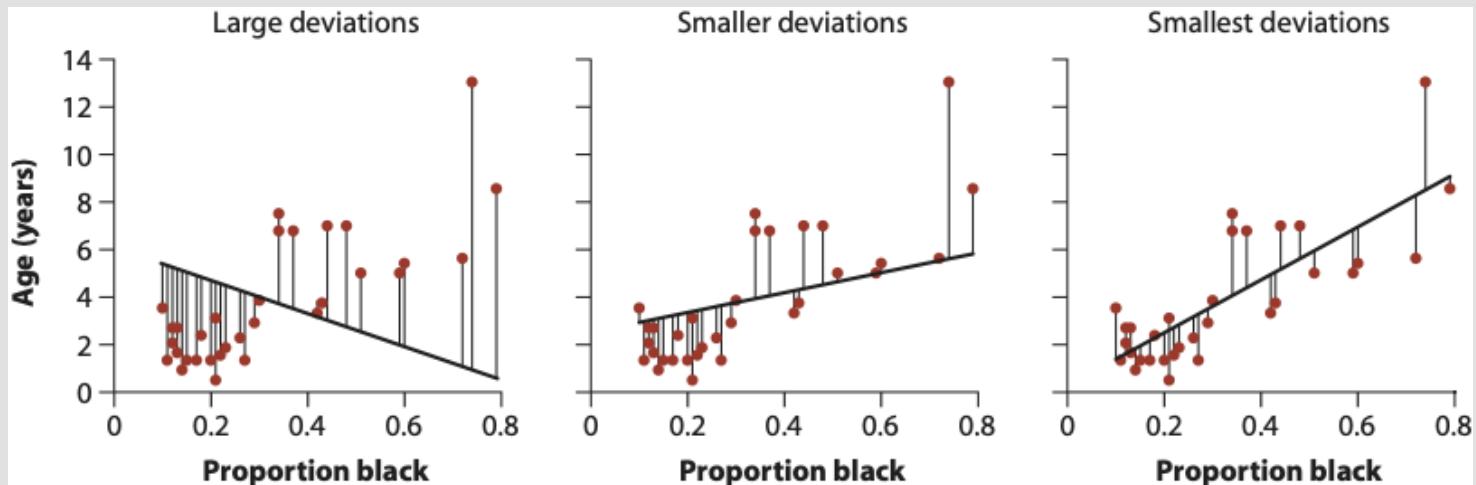
The relationship between the age of a given lion and the proportion of its nose that is black (nose color changes over time!). [Source](#)



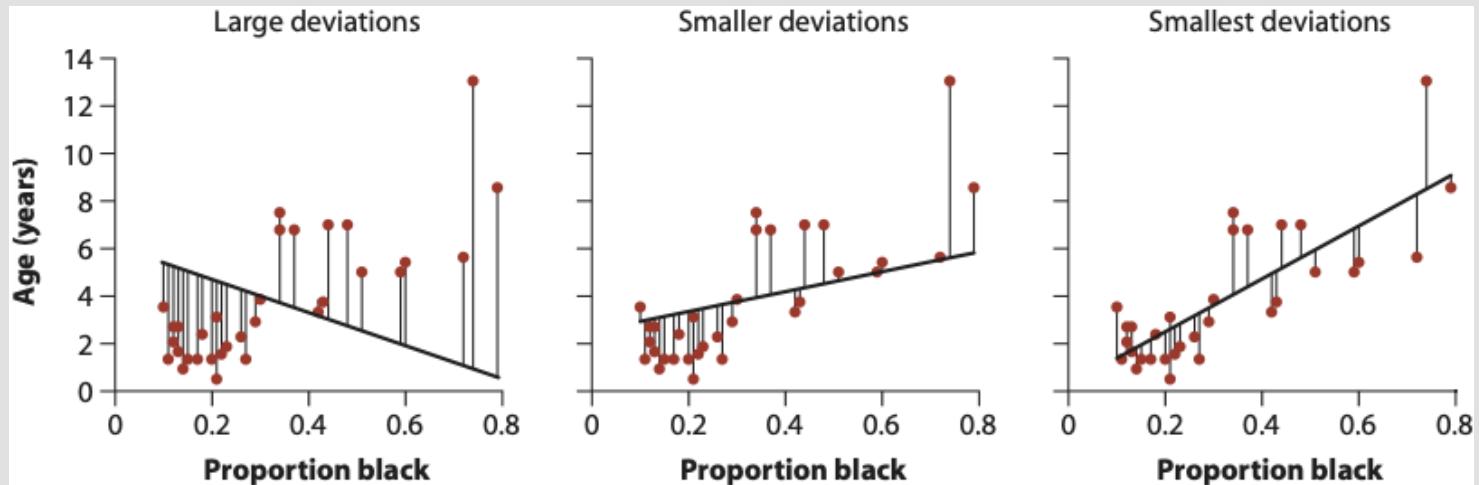
Fitting a model in this case means determining the line-of-best-fit, aka determining the *optimal values* for slope and intercept (*model parameters*)

Residuals: Distance from each point to line-of-best-fit

- Residuals are *errors* - how much does each point deviate (literally, distance) from the *average relationship*? Every point has a residual value.
- Best-fitting line is the line with the *smallest possible Residual Sum of Squares (RSS)*
 - Literally the sum of the squared lengths of each residual line



Focus on the "smallest deviations" panel



- Slope is 10.64
- Intercept is 0.88
- Line is $y = 10.64x + 0.88$. That's our FITTED MODEL!!
 - $Y = 10.64X_1 + 0.88 + \epsilon$
- This model formula says that lions who on average have 50% black noses are on average 6.2 years old.
 - $10.64(0.5) + 0.88 = 6.2$

Null-hypothesis testing and P-values

- P-values are one of the most notoriously misunderstood concepts. They tell you:
 - *Assuming the null hypothesis is true, what is the probability of observing my data?*
- They DO NOT tell you:
 - What is the probability that this result I observe is real?
 - Is the null hypothesis wrong?
 - Is the null hypothesis right?
 - Is the alternative hypothesis wrong?
 - Is the alternative hypothesis right?
 - They really don't tell you much at all, in fact

Null hypotheses are *set in stone*

- Each statistical test you do relies on a highly specific null hypothesis that is *always associated with that statistical test*. There is 0 creativity or wiggle-room.
- In linear models, the null hypotheses are:
 - All $\beta_n = 0$ (coefficients = 0)
 - The $R^2 = 0$
- Each estimated parameter has an associated P-value

Statistical significance is mental gymnastics

Remember: P-values give the probability of observing your data/results *when assuming a TRUE null hypothesis.*

If a P-value is very very small, we say: Gee! That's a small probability! I don't think it's likely that things with low probabilities happen, so maybe actually something else besides the null is going on. *We call this significant.*

If a P-value is not very small, we say: Gee! I think that probabilities that are not very small could totally come to pass. It's not unreasonable to maybe observe this data under the null. *We call this not significant.*

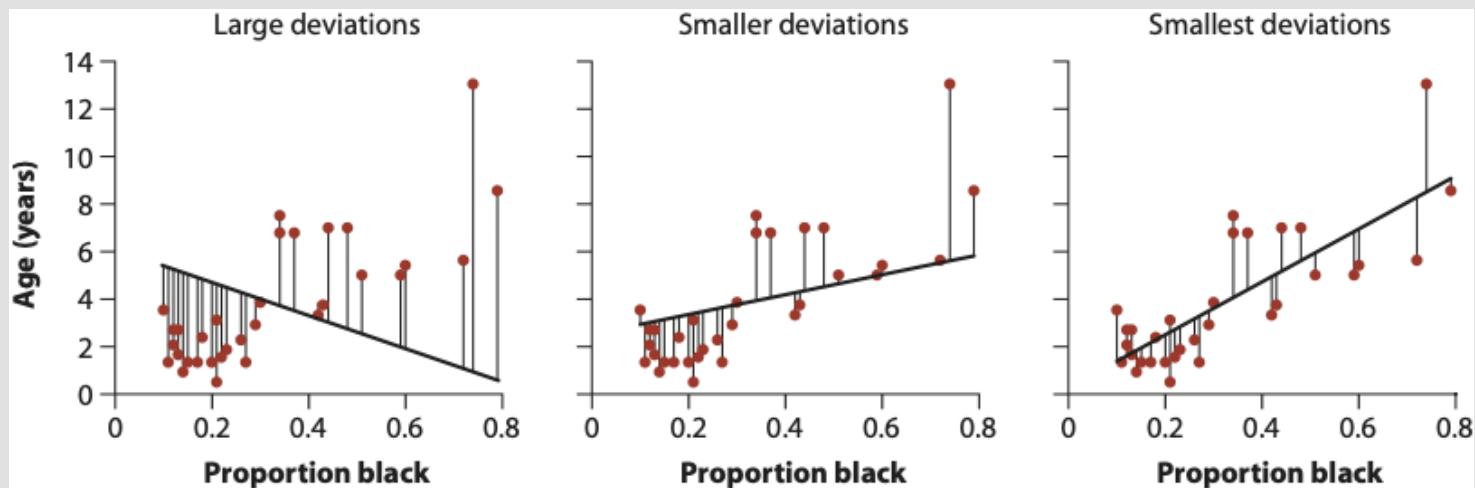
A common threshold for "small" is $P < 0.05$

This number is not special. It is not magic. It's an "historical accident." [See here](#)

"...If one in twenty does not seem high enough odds, we may, if we prefer it, draw the line at one in fifty or one in a hundred.

Personally, the writer prefers to set a low standard of significance at the 5 per cent point, and ignore entirely all results which fails to reach this level. A scientific fact should be regarded as experimentally established only if aproperly designed experiment rarely fails to give this level of significance..." --RA Fisher

Back to our "smallest deviations" lions



Parameter	Estimate	P-value
Slope (a coefficient)	10.64	7.68e-08
Intercept (a coefficient)	0.88	0.133
R^2	0.6113	7.68e-08

Interpreting R^2

- The proportion black variable explains **61.13%** of the variation in lion age *in this dataset*
 - 31.87% of variation is *unexplained*
 - For one of these lions, if I know the proportion black, I know about 61% of what there is to know about how old that lion is
 - If all residuals are 0, then $R^2 = 1$. Aka, if all points precisely are along the regression line, I know entirely what there is to know (100%) about lion age, *in the given dataset*
-
- There are additional methods one can use to determine how good a given PREDICTION is.
 - That model was built from a given dataset, so we only know how well that line-of-best-fit (model!) works for the data at hand (61%). We'd have to test how well this line-of-best-fit works on *other data* to know how good its *predictions* are.

Reminder:

Hypothesis-testing ("science")

- Predictors are based on experimental setup
- *Specific* goal of knowing how those *specific* predictors affect response

Exploratory ("industry"/"big-data")

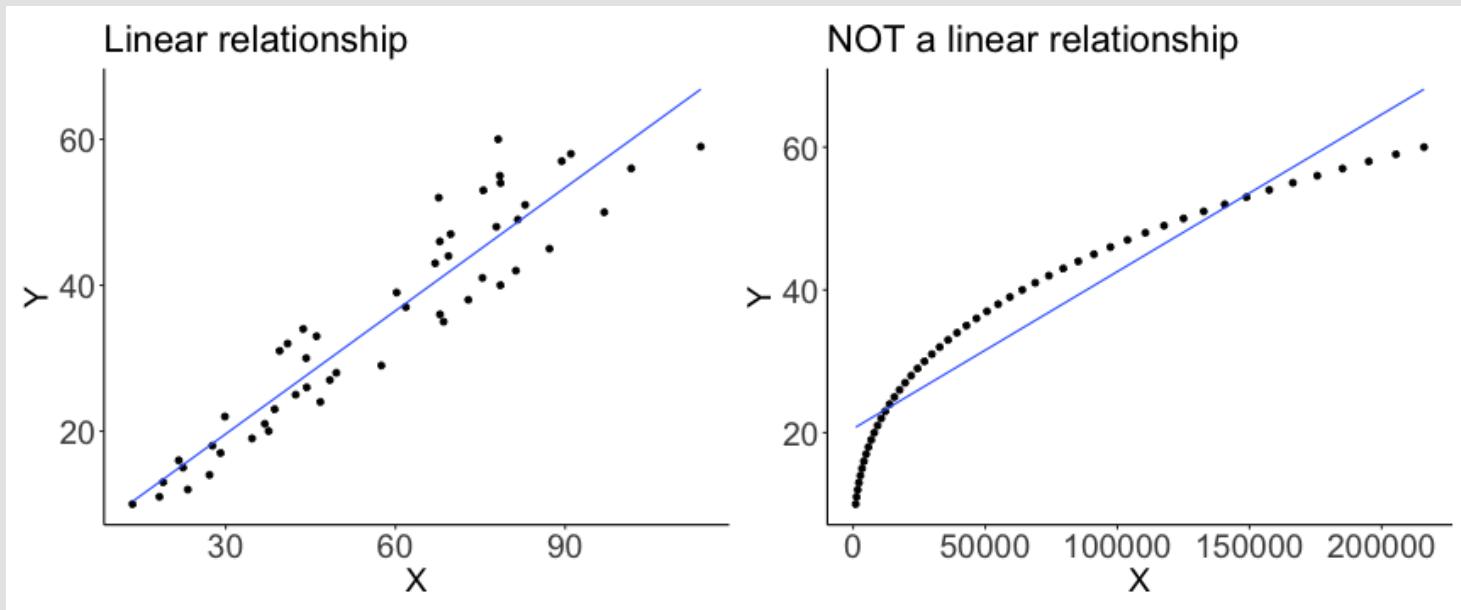
- You have *a bunch* of data and need to figure out, which predictors should I use in my model to best explain the response?
- Less likely you care about specific effects of individual predictors

Linear models ("regression and friends")

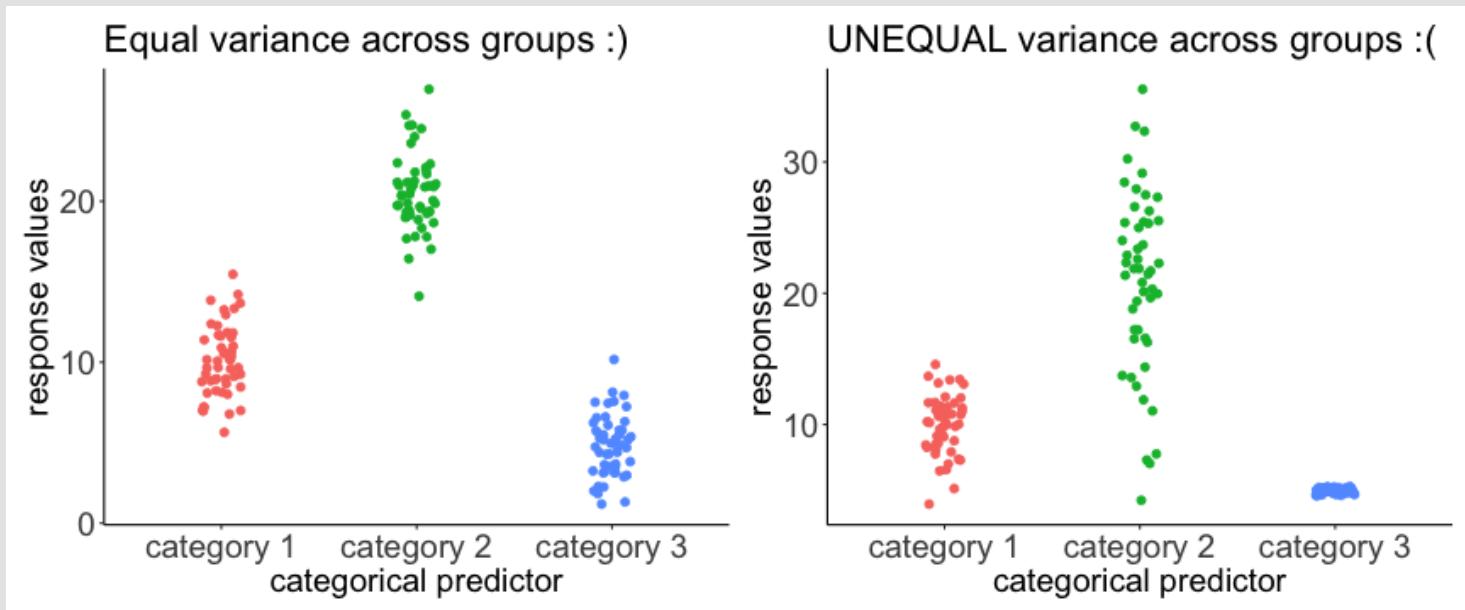
- Explain variation in a *numeric response* variable using *any type and combination of predictors*
- Key assumptions:
 - Any numeric predictors are linearly related to the response
 - The values of the response variable have equal variance across categories of any categorical predictor
 - The *residuals* of the model are normally distributed. There is NO REQUIREMENT for the data itself to follow a normal distribution
 - Predictors should be independent of one another

In this class, we will assume assumptions are met when performing analyses.

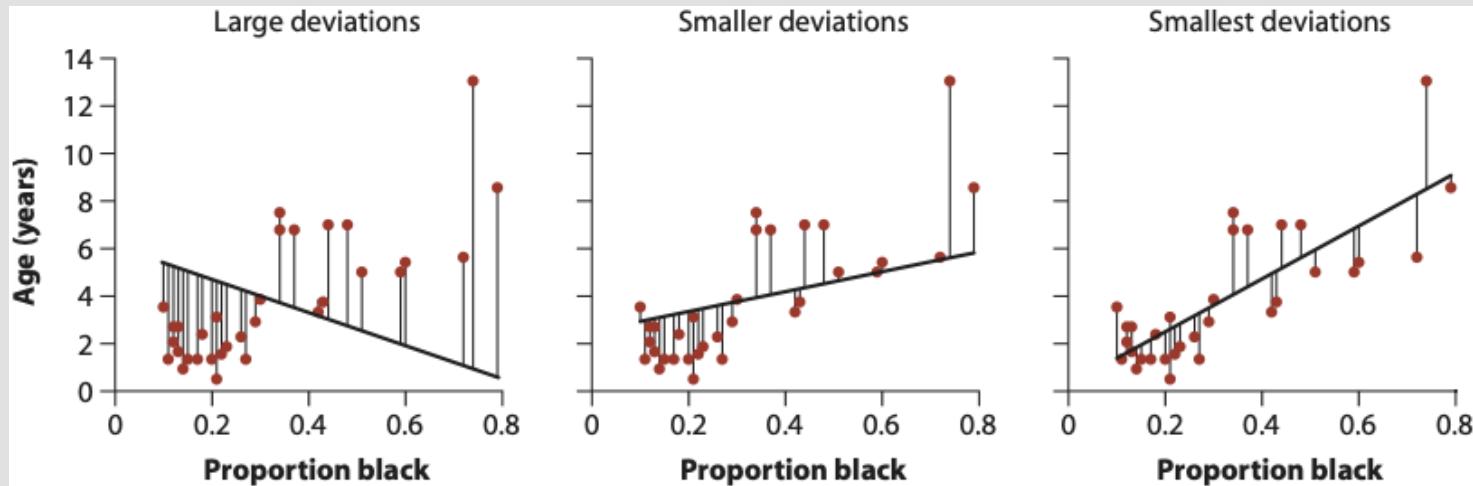
Numeric predictors must be linearly related



Categorical predictors must show equal variance of response



Model *residuals* are normally distributed



Let's build a model

```
library(palmerpenguins) # to access penguins dataset
```

```
penguins
```

```
## # A tibble: 344 x 8
##   species island bill_length_mm bill_depth_mm
##   <fct>   <fct>        <dbl>        <dbl>
## 1 Adelie   Torge...     39.1       18.7
## 2 Adelie   Torge...     39.5       17.4
## 3 Adelie   Torge...     40.3        18
## 4 Adelie   Torge...      NA         NA
## 5 Adelie   Torge...     36.7       19.3
## 6 Adelie   Torge...     39.3       20.6
## 7 Adelie   Torge...     38.9       17.8
## 8 Adelie   Torge...     39.2       19.6
## 9 Adelie   Torge...     34.1       18.1
## 10 Adelie  Torge...     42          20.2
## # ... with 334 more rows, and 4 more variables:
## #   flipper_length_mm <int>, body_mass_g <int>,
## #   sex <fct>, year <int>
```

Make it fit better on slides

```
penguins %>%
  rename(bill_len = bill_length_mm,
         bill_dep = bill_depth_mm,
         flipper = flipper_length_mm,
         mass     = body_mass_g) -> peng

peng

## # A tibble: 344 x 8
##   species island bill_len bill_dep flipper  mass sex
##   <fct>   <fct>    <dbl>    <dbl>    <int> <int> <fct>
## 1 Adelie  Torge...    39.1     18.7     181   3750 male
## 2 Adelie  Torge...    39.5     17.4     186   3800 fema...
## 3 Adelie  Torge...    40.3      18       195   3250 fema...
## 4 Adelie  Torge...     NA       NA       NA     NA <NA>
## 5 Adelie  Torge...    36.7     19.3     193   3450 fema...
## 6 Adelie  Torge...    39.3     20.6     190   3650 male
## 7 Adelie  Torge...    38.9     17.8     181   3625 fema...
## 8 Adelie  Torge...    39.2     19.6     195   4675 male
## 9 Adelie  Torge...    34.1     18.1     193   3475 <NA>
## 10 Adelie  Torge...    42       20.2     190   4250 <NA>
## # ... with 334 more rows, and 1 more variable:
## #   year <int>
```

Linear regression

- To what extent does flipper length explain variation in bill length?
 - Is there a significant relationship between these variables, and if so, how strong is it?
-
- Response variable: `bill_len`
 - Predictor variable: `flipper`

We build linear models with the function `lm()`

```
lm(response ~ predictor(s), data = name_of_dataframe)
```

```
## Model with bill_len response and flipper predictor
fitted_model <- lm(bill_len ~ flipper, data = peng)

## Examine model output ("ugly version")
summary(fitted_model)
```

```
##
## Call:
## lm(formula = bill_len ~ flipper, data = peng)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -8.5792 -2.6715 -0.5721  2.0148 19.1518 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -7.26487   3.20016  -2.27   0.0238 *  
## flipper       0.25477   0.01589   16.03  <2e-16 *** 
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 
## 
## Residual standard error: 4.126 on 340 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.4306,    Adjusted R-squared:  0.4289 
## F-statistic: 257.1 on 1 and 340 DF,  p-value: < 2.2e-16
```

Use the `broom` and `modelr` packages to examine output

```
broom::tidy(fitted_model)
```

```
## # A tibble: 2 x 5
##   term      estimate std.error statistic p.value
##   <chr>      <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept) -7.26     3.20     -2.27 2.38e- 2
## 2 flipper       0.255    0.0159     16.0  1.74e-43
```

```
# Won't give P-value, but if  $R^2 \geq 0.1$ , it's generally significant
modelr::rsquare(fitted_model, peng)
```

```
## [1] 0.430574
```

How about species as a predictor?

- To what extent does species explain variation in bill length?
- Do different species have different mean bill lengths?

```
fitted_model_species <- lm(bill_len ~ species, data = peng)
```

```
## Let's first look at R^2
broom::glance(fitted_model_species) %>%
  select(r.squared, p.value)
```

```
## # A tibble: 1 x 2
##   r.squared  p.value
##       <dbl>    <dbl>
## 1     0.708 2.69e-91
```

The model coefficients are directly related to the means

```
broom::tidy(fitted_model_species)
```

```
## # A tibble: 3 x 5
##   term       estimate std.error statistic p.value
##   <chr>     <dbl>    <dbl>     <dbl>    <dbl>
## 1 (Intercept) 38.8     0.241    161.  2.50e-322
## 2 speciesChinst... 10.0     0.432    23.2  4.23e- 72
## 3 speciesGentoo  8.71     0.360    24.2  5.33e- 76
```

```
peng %>%
  group_by(species) %>%
  summarize(mean_bill_len = mean(bill_len, na.rm=TRUE))
```

```
## # A tibble: 3 x 2
##   species  mean_bill_len
##   <fct>      <dbl>
## 1 Adelie     38.8
## 2 Chinstrap  48.8
## 3 Gentoo    47.5
```

Do you miss the ANOVA table?

Shoutout to how much you loved or will love this topic in Biometry!

```
as_anova <- aov(fitted_model_species)
summary(as_anova)

##           Df Sum Sq Mean Sq F value Pr(>F)
## species      2    7194     3597   410.6 <2e-16 ***
## Residuals   339    2970        9
## ---
## Signif. codes:
## 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 2 observations deleted due to missingness
```

How about both?

```
fitted_model_both <- lm(bill_len ~ flipper + species, data = peng)
```

This model *controls for variation across species*. It asks:

- Controlling for species, to what extent does flipper length explain variation in bill length?
- Is there a relationship between flipper length and bill length, when controlling for species?
- Controlling for flipper length, to what extent does the species explain variation in bill length?
- Do different species have significantly different bill lengths, when controlling for flipper length?

This model explains ~77.6% of variation in bill length

- In this dataset, if we know the flipper length and species, we know ~77.6% of what there is to know about bill length
- With multiple predictors, it's challenging to interpret coefficients. We're going to focus on R^2 in these circumstances.

```
rsquare(fitted_model_both, peng)
```

```
## [1] 0.775847
```

Interaction effects

Is the bill_len/flipper relationship the same or different across species? Look at slopes.

```
ggplot(peng) +  
  aes(x = flipper, y = bill_dep, color = species) +  
  geom_point() +  
  geom_smooth(method = "lm") +  
  theme(legend.position = "bottom")
```

Testing for interaction effects

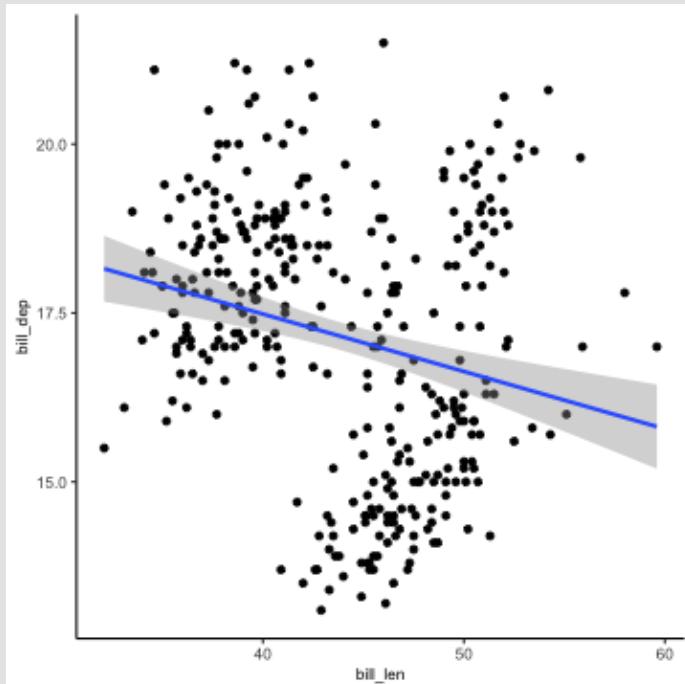
```
fitted_model_both_int <- lm(bill_len ~ flipper * species, data = peng)

tidy(fitted_model_both_int)
```

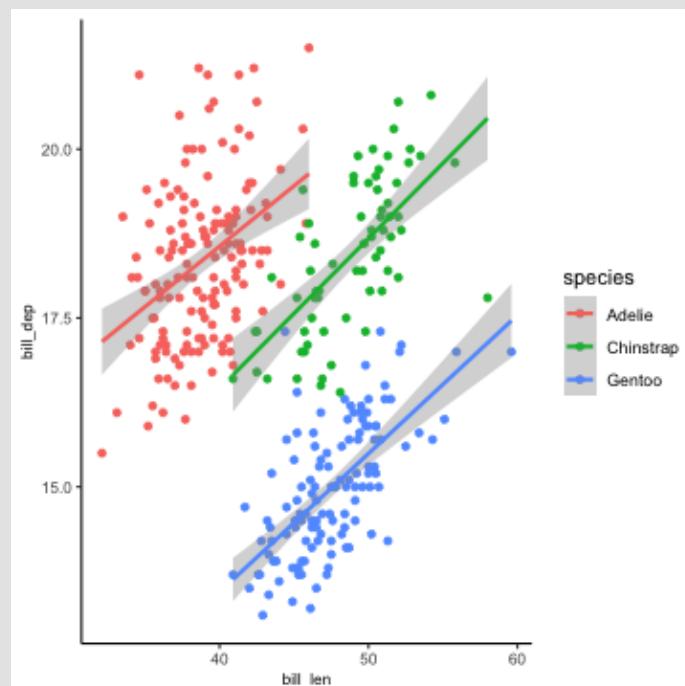
```
## # A tibble: 6 x 5
##   term            estimate std.error statistic p.value
##   <chr>          <dbl>     <dbl>      <dbl>    <dbl>
## 1 (Intercept)    13.6      6.05      2.25    2.54e-2
## 2 flipper        0.133     0.0318     4.17    3.91e-5
## 3 speciesChinstr... -7.99     10.5      -0.763   4.46e-1
## 4 speciesGentoo   -34.3     9.82      -3.50    5.37e-4
## 5 flipper:specie...  0.0881    0.0540     1.63    1.04e-1
## 6 flipper:specie...  0.182     0.0478     3.80    1.71e-4
```

You MUST control for *confounding factors*

```
ggplot(peng) +  
  aes(x = bill_len, y = bill_dep) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

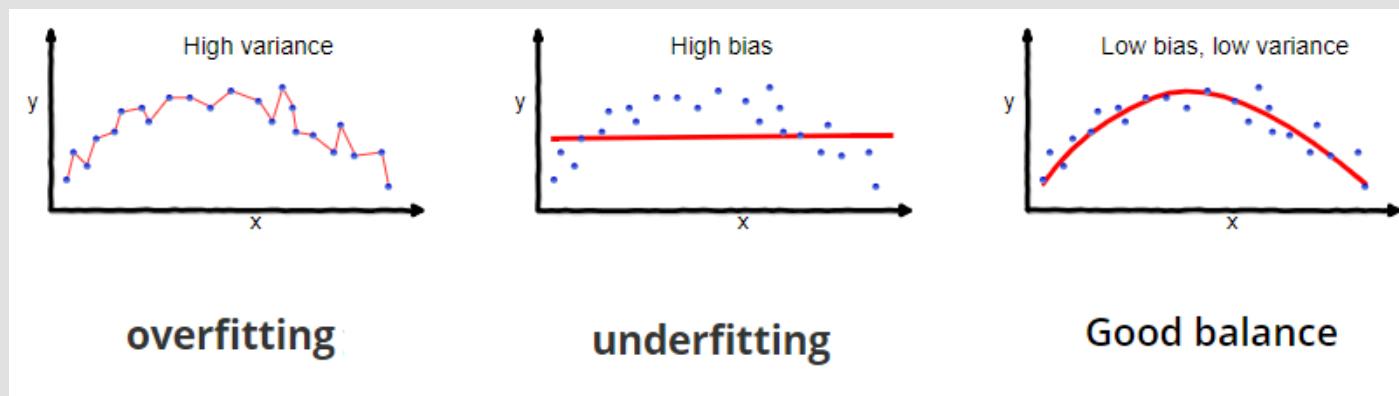


```
ggplot(peng) +  
  aes(x = bill_len, y = bill_dep,  
      color = species) +  
  geom_point() +  
  geom_smooth(method = "lm")
```



Into the Machine Learning World

- Need to determine optimal predictors along the **bias/variance trade-off**.
There are many ways to do this.



- Fit model
- Run diagnostics to ask whether and/or how badly assumptions were violated
- Run diagnostics to ask how well the model performs *on data it has never seen*

Choosing predictors with the `step()` function

- This is called **model selection**: We are selecting which predictors to put into our model, using "fancy statistics."

```
peng_nona <- peng %>% drop_na() # step() gets irritated with NAs  
  
## predictor as period . means: use all!!  
full_model <- lm(bill_len ~ ., data = peng_nona)  
  
## Use step() function to pick which predictors are worth keeping  
## trace = FALSE means less annoying output, trust me...  
final_model <- step(full_model, trace=FALSE)
```

Let's compare the full vs. final model

```
## Full model: All columns (except bill length) are predictors
tidy(full_model) %>% pull(term)
## [1] "(Intercept)"      "speciesChinstrap"
## [3] "speciesGentoo"    "islandDream"
## [5] "islandTorgersen"   "bill_dep"
## [7] "flipper"           "mass"
## [9] "sexmale"           "year"

## Quickly determine the R^2 for the model
rsquare(full_model, peng_nona)
## [1] 0.8400442
```

```
## Final model: Predictors chosen by step()
tidy(final_model) %>% pull(term)
## [1] "(Intercept)"      "speciesChinstrap"
## [3] "speciesGentoo"    "bill_dep"
## [5] "flipper"           "mass"
## [7] "sexmale"

rsquare(final_model, peng_nona)
## [1] 0.8385834
```

Gauging error in our model with RMSE

- RMSE (Root Mean Square Error) is the standard deviation of model residuals, in *units of the response*
 - Model residuals represent *error*: what variation in the data is NOT captured by the model?
 - High RMSE = high error spread. Points are generally more spread out from the line-of-best-fit
 - Low RMSE = low error spread. Points are generally closer to the line-of-best-fit
- RMSE is one of many metrics used to gauge model performance, but RMSE and R^2 are among the most commonly-used

```
## Quickly determine the RMSE for the model
modelr::rmse(final_model, peng_nona)
## [1] 2.193828

## Peek at bill lengths for context!
summary(peng_nona$bill_len)
##      Min. 1st Qu. Median      Mean 3rd Qu.      Max.
## 32.10    39.50   44.50    43.99   48.60    59.60
```

Predicting future outcomes

- Recall, our final model to explain bill length has these predictors:
 - species, bill_dep, flipper, mass, sex
- I made a new penguin friend with these characteristics:
 - Penguin is a male Chinstrap
 - Bill depth is 18.5
 - Flipper length is 211
 - Mass is 5200
- What does the model think its bill length is? **Predict = run the model on new data.**

Using the `predict()` function:

1. Make a tibble with new data. Must match original data naming/spelling!
2. Use function as `predict(fitted model, tibble with new data)`

First, make the tibble for prediction

```
head(peng_nona)
```

```
## # A tibble: 6 x 8
##   species island bill_len bill_dep flipper  mass sex
##   <fct>   <fct>     <dbl>     <dbl>    <int> <int> <fct>
## 1 Adelie   Torge...    39.1     18.7     181   3750 male 
## 2 Adelie   Torge...    39.5     17.4     186   3800 fema...
## 3 Adelie   Torge...    40.3      18       195   3250 fema...
## 4 Adelie   Torge...    36.7     19.3     193   3450 fema...
## 5 Adelie   Torge...    39.3     20.6     190   3650 male 
## 6 Adelie   Torge...    38.9     17.8     181   3625 fema...
## # ... with 1 more variable: year <int>
```

```
new_penguin <- tibble(
  species = "Chinstrap",
  sex      = "male",
  bill_dep = 18.5,
  flipper  = 223,
  mass     = 5200
)
new_penguin
```

```
## # A tibble: 1 x 5
##   species   sex   bill_dep flipper  mass
##   <chr>     <chr>     <dbl>    <dbl> <dbl>
## 1 Chinstrap male     18.5     223   5200
```

Second, predict!

```
predict(final_model, new_penguin)
```

```
##           1  
## 53.32675
```

Recall our model's RMSE:

```
rmse(final_model, peng_nona)
```

```
## [1] 2.193828
```

We expect, on average, predictions will be ~2.19 units wrong. But how wrong is *this specific* prediction?

new_penguin

```
## # A tibble: 1 x 5
##   species    sex bill_dep flipper  mass
##   <chr>      <chr>    <dbl>    <dbl>  <dbl>
## 1 Chinstrap male     18.5    223 5200
```

```
peng_nona %>%  
  select(species, sex, bill_dep, flipper, mass) -> peng_nona_preds  
summary(peng_nona_preds)
```

```

##           species      sex    bill_dep
## Adelie     :146 female:165   Min.   :13.10
## Chinstrap: 68  male  :168   1st Qu.:15.60
## Gentoo    :119                    Median :17.30
##                               Mean   :17.16
##                               3rd Qu.:18.70
##                               Max.   :21.50
##           flipper      mass
## Min.   :172   Min.   :2700
## 1st Qu.:190   1st Qu.:3550
## Median :197   Median :4050
## Mean   :201   Mean   :4207
## 3rd Qu.:213   3rd Qu.:4775
## Max.   :231   Max.   :6300

```

```
new_penguin
```

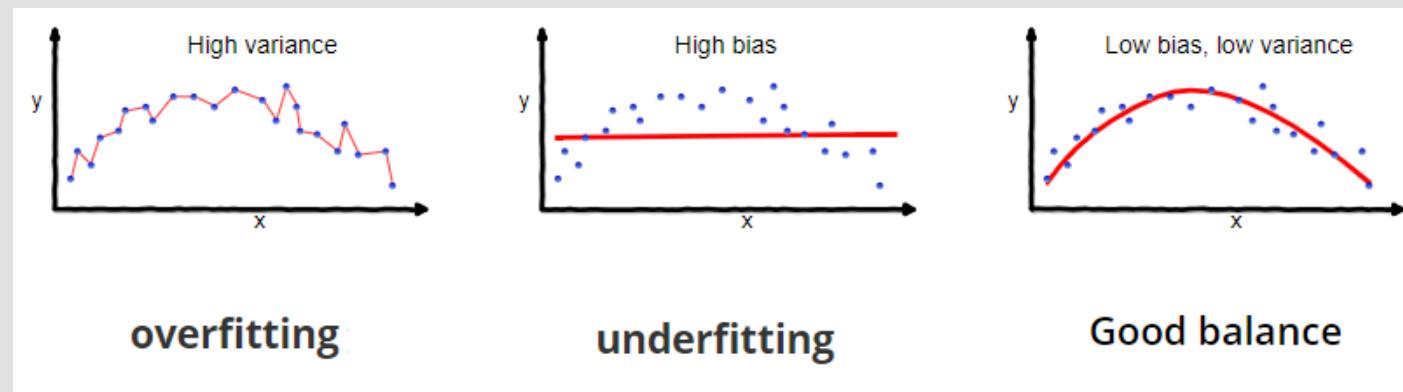
```
## # A tibble: 1 x 5
##   species   sex bill_dep flipper  mass
##   <chr>     <chr>    <dbl>    <dbl> <dbl>
## 1 Chinstrap male      18.5     223  5200
```

```
peng_nona_preds %>%
  filter(sex == "male", species == "Chinstrap") %>%
  select(-sex, -species) -> peng_nona_preds2
summary(peng_nona_preds2)
```

```
##       bill_dep        flipper         mass
##   Min.   :17.50   Min.   :187.0   Min.   :3250
##   1st Qu.:18.80   1st Qu.:196.0   1st Qu.:3731
##   Median :19.30   Median :200.5   Median :3950
##   Mean   :19.25   Mean   :199.9   Mean   :3939
##   3rd Qu.:19.80   3rd Qu.:203.0   3rd Qu.:4100
##   Max.   :20.80   Max.   :212.0   Max.   :4800
```

It is important to assess how dependent your model is on initial data

- We **train** models on a given dataset
- The properties of that **training dataset** influence how the model will perform
 - If I build a model solely using Gentoo penguin data, the model will probably be unhelpful for studying Chinstrap penguins
- Models that work well on training data but perform terribly on new data are usually **overfit**
 - Model is "overly-tailored" to training data
 - Aka, $R^2 = 0.9$ on training data, but predictions are *really really WRONG* when running model on new data



Performing model *validation* (aka *evaluation*)

- Asks: How good is the model at explaining variation **in data it does NOT know about?** Should we even bother using our model to predict future outcomes?

Industry really wants to know the answer to this. Academic science is usually less concerned (but should we be more concerned? maybe!)

1. Determine which predictors you will use
 - In this class, we'll do this using the `step()` function (not "hypothesis-testing")
2. Run a cross-validation procedure to gauge model performance on new data

Cross-validation with testing and training groups

- Randomly **split** your dataset into two parts:
 - The *training split* (usually 60-80% of the data) **builds** aka **trains** the model

The full procedure to predict bill depth

Step 1: Determine which predictors your model will use

You either know this information already, or you need to use model selection:

```
peng_nona <- peng %>% drop_na() # reminder, step() gets irritated with NAs
fit <- step( lm(bill_len ~ ., data = peng_nona), trace=FALSE)

# Grab the model formula inside fit to tell us the predictors
# Useful to grab this if you need to use step()
fit_formula <- fit$call$formula
fit_formula

## bill_len ~ species + bill_dep + flipper + mass + sex
```

Step 2: Divide data into testing and training

```
# Choose your favorite number for reproducibility. !! SO IMPORTANT!!
set.seed(1)
```

```
training_size <- 0.6
peng_nona %>%
  # Sample random fraction of rows with this dplyr function
  sample_frac(training_size) -> peng_training
nrow(peng_training)
```

```
## [1] 200
```

```
peng_testing <- anti_join(peng_nona, peng_training)
nrow(peng_testing)
```

```
## [1] 133
```

Step 3: Fit model formula using the training data

```
training_fit <- lm(fit_formula, data = peng_training)
```

Step 4: Ask how well *model built from training data* performs on the *training AND the testing data*

The `modelr` package makes this super easy :)

```
# Training data: The model "has seen" this data
rsquare(training_fit, peng_training)
## [1] 0.8588903
rmse(training_fit, peng_training)
## [1] 1.962553
```

```
# Testing data: The model "has never seen" this data
rsquare(training_fit, peng_testing)
## [1] 0.8107401
rmse(training_fit, peng_testing)
## [1] 2.522255
```

Step 4: Draw *qualitative* conclusions

Data	R^2	RMSE
Training	0.86	1.96
Testing	0.81	2.522

The model performed pretty well on the testing data! Our validation procedure shows the model is not overfit. Predictions are going to be pretty decent.

This would be super concerning:

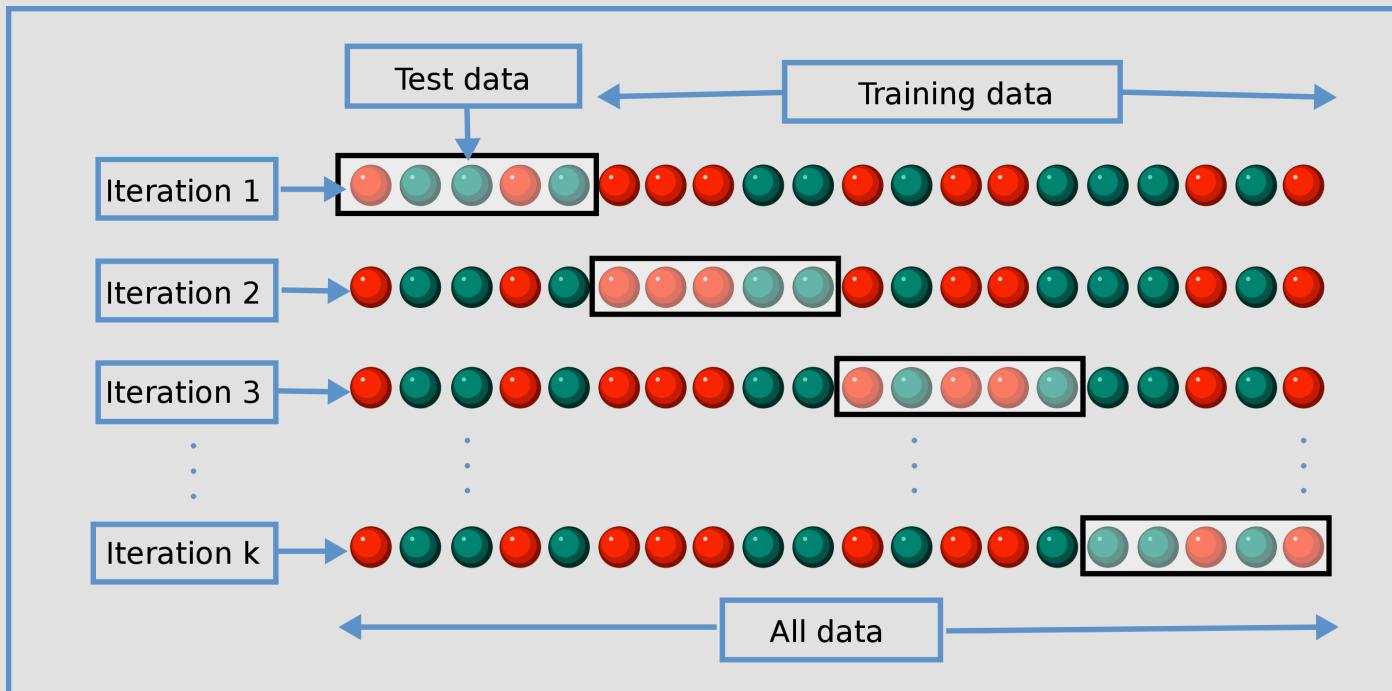
Data	R^2	RMSE
Training	0.86	1.96
Testing	0.54	22.4

```
summary(peng_nona$bill_len)
```

```
##      Min. 1st Qu. Median      Mean 3rd Qu.   Max.
## 32.10   39.50  44.50  43.99  48.60  59.00
```

Another way: K-fold cross validation

- Randomly divide the whole dataset into "K" equal chunks aka folds
- Perform K iterations of model training and testing
 - "Hold back" data each time for testing!
- Get RMSE and R^2 for each iteration, and look at full distribution



Another way: Leave-one-out cross validation (LOOCV)

- K-folds on speed: each "test" size is $N=1!!$
- For small datasets, LOOCV probably "better"

