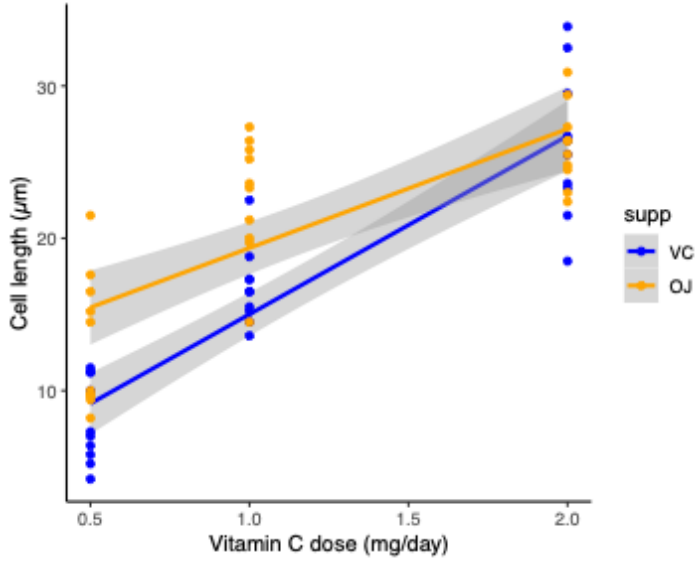
RandomForests

Starting point:

- Clarity about what is your response variable and what are explanatory variables
- Clarity about how you expect your response variable to be distributed
- Scientific questions about the relationship between response and explanatory variables
- Data in a long-form dataframe or tibble

Linear models can’t do everything

- Get very complicated with interactions among more than three fixed effects
  - Interactions in $y \sim x_1 * x_2 * x_3$ is hard enough to interpret
  - what if have hundreds of explanatory variables? (e.g. ‘omics)
- Overly complex if just want to make a prediction on new data
  - May not want parameters or their errors
  - e.g. if just need best guess at a categorical response
- Data may not have a well defined response distribution
  - Whether normal or otherwise
- Response and explanatory variables may not have clear parametric relationships
  - Whether linear or nonlinear, which is possible too



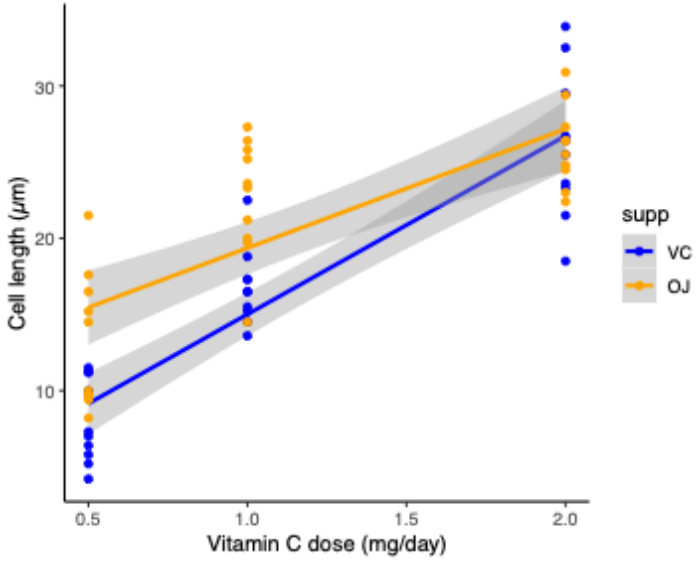
Linear modelling is surprisingly robust to failing to meet assumptions,
But sometimes need something else

Linear models can't do everything

- Get very complicated with interactions among more than three fixed effects
  - Interactions in $y \sim x_1 * x_2 * x_3$ is hard enough to interpret
  - what if have hundreds of explanatory variables? (e.g. 'omics)

- Overly complex if just want to make a prediction on new data
  - May not want parameters or their errors
  - e.g. if just need best guess at a categorical response

- Data may not have a well defined response distribution
  - Whether normal or otherwise



- Response and explanatory variables may not have clear parametric relationships
  - Whether linear or nonlinear, which is possible too

Linear modelling is surprisingly robust to failing to meet assumptions,
But sometimes need something else

- Machine learning
  - Wide range of tools and approaches, usually without some of the issues above
  - Different tradition from computer science not statistics – 'Statistics without the proofs'
  - Practically often 'just another model' implemented in R, similar to linear models
  - Not without assumptions, pitfalls or limitations, but different!

Linear models can’t do everything

- Machine learning
  - Wide range of tools and approaches, usually without some of the issues above
  - Different tradition from computer science not statistics – ‘Statistics without the proofs’
  - Practically often ‘just another model’ implemented in R, similar to linear models
  - Not without assumptions, pitfalls or limitations, but different!

Linear Models	Random Forests
Usually single response	Usually single response
Usually limited number of independent explanatory variables	Many explanatory variables
Limited numbers of pre-defined interactions possible	Very flexible interactions among large numbers of variables
Strong assumptions (linearity, independence, homoscedasticity, error distribution)	Limited assumptions about data, can be continuous, categorical, outliers ok
Very explicit interpretation of parameters	Hard to interpret workings of model
Hypothesis testing strong	Hypothesis testing weak

Linear models can't do everything

- Machine learning
  - Wide range of tools and approaches, usually without some of the issues above
  - Different tradition from computer science not statistics – 'Statistics without the proofs'
  - Practically often 'just another model' implemented in R, similar to linear models
  - Not without assumptions, pitfalls or limitations, but different!

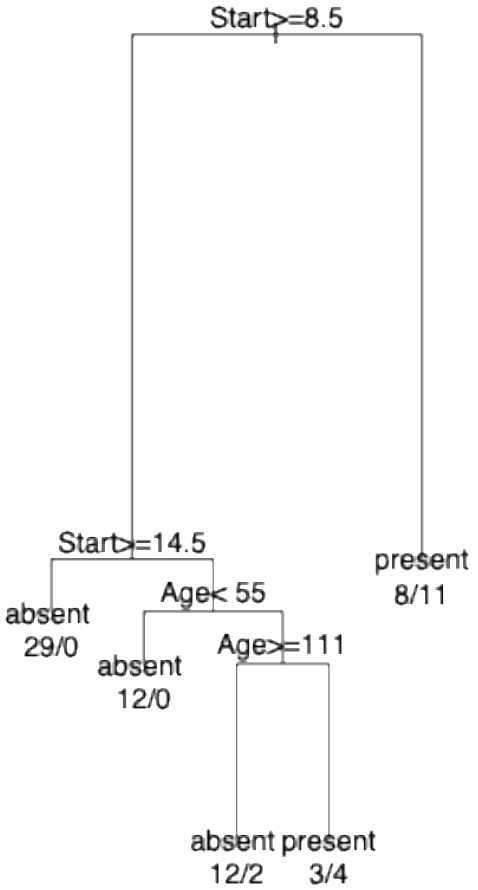
Linear Models	Random Forests
Usually single response	Usually single response
Usually limited number of independent explanatory variables	Many explanatory variables
Limited numbers of pre-defined interactions possible	Very flexible interactions among large numbers of variables
Strong assumptions (linearity, independence, homoscedasticity, error distribution)	Limited assumptions about data, can be continuous, categorical, outliers ok
Very explicit interpretation of parameters	Hard to interpret workings of model
Hypothesis testing strong	Hypothesis testing weak

"If our goal as a field is to use data to solve problems, then we need to move away from exclusive dependence on data models and adopt a more diverse set of tools."

Breiman, L. **Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author)**. *Statistical Science* 16 (2001). 10.1214/ss/1009213726

What is a random forest?

- Machine learning
  - Wide range of tools and approaches, usually without some of the issues above
  - Different tradition from computer science not statistics – 'Statistics without the proofs'
  - Practically often 'just another model' implemented in R, similar to linear models
  - Not without assumptions, pitfalls or limitations, but different!
- Tree-based method
  - Split response variable into two on the basis of an explanatory variable
  - Pick the split that minimizes the variability of the response in the subsets produced
  - Keep splitting until you run out of data! (or come up against a stopping criterion)
    - Really efficient algorithms to do this
- Kyphosis data
  - Children who have had Corrective Spinal Surgery
  - What determines if they end up with a curved spine (kyphosis – absent/present)?
  - Explanatory variables: Start (top vertebra operated on), Age (in months)



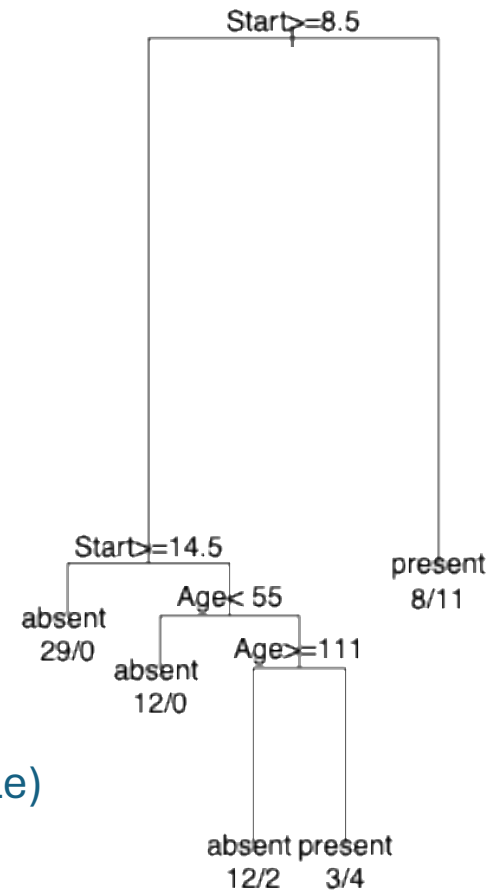
- Very flexible
  - Would work exactly the same if response was >2 categories (e.g. none, some, lots), or continuous (curvature angle)
- Can be extended beyond one tree
  - Different (random selection of) Explanatory variables
  - Different (random) subsets of data
  - Combine with a voting system (for which data point goes to which outcome)
  - Do for hundreds or thousands of trees – get random forests™

Breiman, L. **Random forests**. *Machine Learning* 45, 5-32 (2001). 10.1023/A:1010933404324

Fitting a random forest

- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
> penguins
# A tibble: 344 × 8
   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g sex      year
   <fct>   <fct>              <dbl>         <dbl>             <int>       <int> <fct>   <int>
 1 Adelie  Torgersen           39.1          18.7               181        3750 male    2007
 2 Adelie  Torgersen           39.5          17.4               186        3800 female  2007
 3 Adelie  Torgersen           40.3          18                 195        3250 female  2007
 4 Adelie  Torgersen           NA            NA                  NA          NA NA      2007
 5 Adelie  Torgersen           36.7          19.3               193        3450 female  2007
 6 Adelie  Torgersen           39.3          20.6               190        3650 male    2007
 7 Adelie  Torgersen           38.9          17.8               181        3625 female  2007
 8 Adelie  Torgersen           39.2          19.6               195        4675 male    2007
 9 Adelie  Torgersen           34.1          18.1               193        3475 NA      2007
10 Adelie  Torgersen           42            20.2               190        4250 NA      2007
# ℹ 334 more rows
```



- ‘Only’ 7 variables in addition to species
  - Would be pretty complicated to fit a linear model to predict species (multinomial variable)
  - Don’t care about specific parameters, just want to be able to be sure of species from measurements – is it possible?

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```

- Very similar syntax to linear models
  - dot represents ‘all the other variables’
  - Need to get rid of missing values somehow (see also na.roughfix and rfImpute)

Understanding a random forest

- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```

```
> rf <- randomForest(species ~ ., data = penguins |> na.omit())
> rf

Call:
 randomForest(formula = species ~ ., data = na.omit(penguins))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 1.2%
Confusion matrix:
          Adelie Chinstrap Gentoo class.error
Adelie       144         2      0  0.01369863
Chinstrap      2        66      0  0.02941176
Gentoo         0         0    119  0.00000000
```

- OOB = “out of bag”
  - i.e. based on data not directly used to build the tree
  - Shouldn’t be ‘**over-fitted**’

Understanding a random forest

- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```

```
> rf <- randomForest(species ~ ., data = penguins |> na.omit())
> rf

Call:
 randomForest(formula = species ~ ., data = na.omit(penguins))
               Type of random forest: classification
                     Number of trees: 500
No. of variables tried at each split: 2

        OOB estimate of  error rate: 1.2%
Confusion matrix:
          Adelie Chinstrap Gentoo class.error
Adelie       144         2      0  0.01369863
Chinstrap      2        66      0  0.02941176
Gentoo         0         0    119  0.00000000
```

- OOB = “out of bag”
  - i.e. based on data not directly used to build the tree
  - Shouldn’t be ‘**over-fitted**’
  - 1.2% pretty low for an error rate
- Sometimes mixes up Chinstrap and Adelie
  - Usually doesn’t
  - Always gets Gentoo
- Seems pretty good, should we believe it?
  - Split into training and test sets
  - Fit model on training set, test how model works on test set

Understanding a random forest

- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```

```
> set.seed(1)
> train <- penguins |>
+   na.omit() |>
+   slice_sample(prop = 0.8) #randomly sample 80% of rows

> test <- penguins |>
+     na.omit() |>
+   anti_join(train) #use the remaining 20% of rows for a test set
Joining with `by = join_by(species, island, bill_length_mm, bill_depth_mm, flipper_length_mm,
body_mass_g, sex, year)`

>  rf_0.8 <- randomForest(species ~ ., data = train)

> test_pred <- predict(rf_0.8, newdata = test)  # predict species for test rows
> table(Actual = test$species, Predicted = test_pred) # confusion matrix
           Predicted
Actual      Adelie Chinstrap Gentoo
  Adelie        31         0      0
  Chinstrap      1        10      0
  Gentoo         0         0     25
> mean(test_pred != test$species) #what proportion of the time is there an error
[1] 0.01492537
```

- OOB = “out of bag”
  - i.e. based on data not directly used to build the tree
  - Shouldn’t be ‘**over-fitted**’
  - 1.2% pretty low for an error rate
- Sometimes mixes up Chinstrap and Adelie
  - Usually doesn’t
  - Always gets Gentoo
- Seems pretty good, should we believe it?
  - Split into training and test sets
  - Fit model on training set, test how model works on test set

Understanding a random forest

- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```

```
> set.seed(1)
> train <- penguins |>
+   na.omit() |>
+   slice_sample(prop = 0.8) #randomly sample 80% of rows
>
> test <- penguins |>
+     na.omit() |>
+   anti_join(train) #use the remainina 20% of rows for a test set
Joining with `by = join_by(species, island, bill_length_mm, bill_depth_mm, flipper_length_mm,
body_mass_g, sex, year)`
>
>  rf_0.8 <- randomForest(species ~ ., data = train)
>
> test_pred <- predict(rf_0.8, newdata = test)  # predict species for test rows
> table(Actual = test$species, Predicted = test_pred) # confusion matrix
           Predicted
Actual      Adelie Chinstrap Gentoo
  Adelie        31         0      0
  Chinstrap      1        10      0
  Gentoo         0         0     25
> mean(test_pred != test$species) #what proportion of the time is there an error
[1] 0.01492537
```

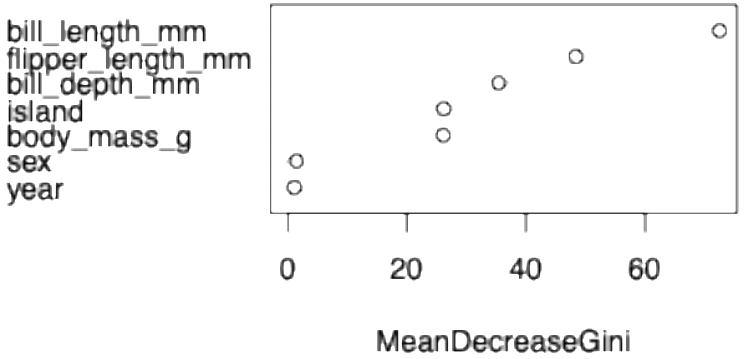
- OOB = “out of bag”
  - i.e. based on data not directly used to build the tree
  - Shouldn’t be ‘**over-fitted**’
  - 1.2% pretty low for an error rate
- Sometimes mixes up Chinstrap and Adelie
  - Usually doesn’t
  - Always gets Gentoo
- Seems pretty good, should we believe it?
  - Split into training and test sets
  - Fit model on training set, test how model works on test set

Understanding a random forest

- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```

```
> set.seed(1)
> train <- penguins |>
+   na.omit() |>
+   slice_sample(prop = 0.8) #randomly sample 80% of rows
>
> test <- penguins |>
+     na.omit() |>
+   anti_join(train) #use the remaining 20% of rows for a test set
Joining with `by = join_by(species, island, bill_length_mm, bill_depth_mm, flipper_length_mm,
body_mass_g, sex, year)`
>
>  rf_0.8 <- randomForest(species ~ ., data = train)
>
> test_pred <- predict(rf_0.8, newdata = test)  # predict species for test rows
> table(Actual = test$species, Predicted = test_pred) # confusion matrix
           Predicted
Actual      Adelie Chinstrap Gentoo
  Adelie        31         0      0
  Chinstrap      1        10      0
  Gentoo         0         0     25
> mean(test_pred != test$species) #what proportion of the time is there an error
[1] 0.01492537
```



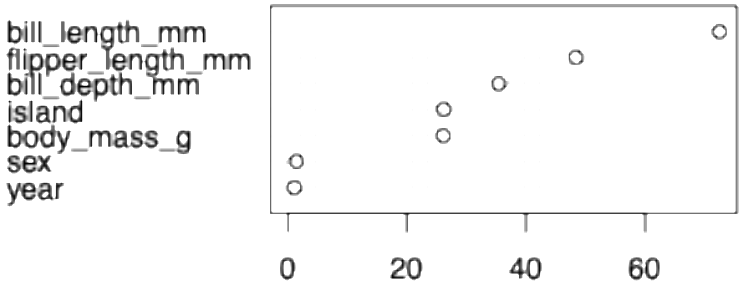

```
varImpPlot(rf)
```

- OOB = “out of bag”
  - i.e. based on data not directly used to build the tree
  - Shouldn’t be ‘**over-fitted**’
  - 1.2% pretty low for an error rate
- Sometimes mixes up Chinstrap and Adelie
  - Usually doesn’t
  - Always gets Gentoo
- Seems pretty good, should we believe it?
  - Split into training and test sets
  - Fit model on training set, test how model works on test set
- 1.5% error (only one actual misclassification)
  - Still very low – is doing a good job
- Bill length most important for determining species
  - Reassuring that sex and year aren’t important
  - Are other/better measures of importance

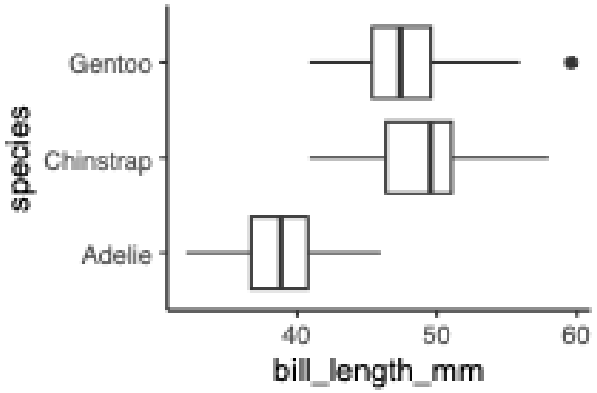
Understanding a random forest

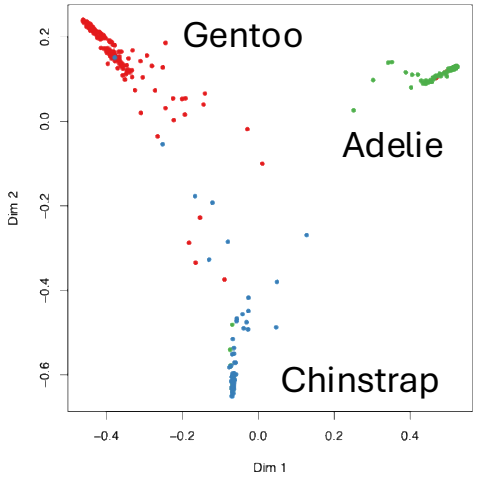
- Back to the penguins
  - Fit a model to decide which species a penguin is based on its measurements

```
rf <- randomForest(species ~ ., data = penguins |> na.omit())
```




```
varImpPlot(rf)
```





- Bill length on its own doesn't get far
  - Can't really see how it's working
  - **Black box** model
  - Different ways of looking deeper: Boruta, randomForestExplainer, pdp…

- OOB = "out of bag"
  - i.e. based on data not directly used to build the tree
  - Shouldn't be '**over-fitted**'
  - 1.2% pretty low for an error rate
- Sometimes mixes up Chinstrap and Adelie
  - Usually doesn't
  - Always gets Gentoo
- Seems pretty good, should we believe it?
  - Split into training and test sets
  - Fit model on training set, test how model works on test set
- 1.5% error (only one actual misclassification)
  - Still very low – is doing a good job
- Bill length most important for determining species
  - Reassuring that sex and year aren't important
  - Are other/better measures of importance