Lab 2: Independant Contrasts

Comparative Biology and Macroevolution

April 12, 2019

Classwork

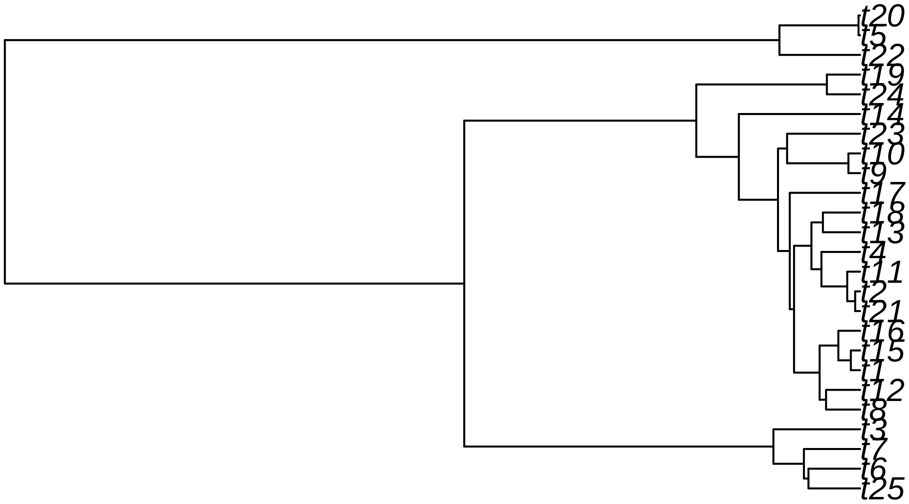
One kind of analysis that is frequently done by evolutionary biologist is trait correlations. For example, we might ask whether brain size and body size are correlated, or if the cribiform plate surface area correlates to the number of olfactory receptor genes. While the analysis you may be more familiar with deals with comparing individuals from the same species, in this class, we will be thinking of trait correlations betweeen different species.

One problem with a naive analysis of trait data between species is that species are not independent. In his 1985 paper, Felsenstein points out that species are part of a “hierarchically stuctured phylogeny” and therefore not independent. Since this insight, there have been multiple approaches that have been developed to help mitigate this problem. In this lab, we will be using the first *fully* phylogenetic method, termed by Feslsenstein as independent contrasts (IC), to incorporate a phylogenetic perspective on comparative data.

Import Data

```
library(ape)
```

```
tree <- read.tree("lab2_tree_new.tre")
plot(tree)
```



```
traits <- read.csv("lab2_data1.csv", stringsAsFactors = F)
traits
```

```
##    Taxon   trait1   trait2
## 1     t1 18.32059 40.62082
## 2    t10 17.58344 41.70659
## 3    t11 18.45661 40.78300
## 4    t12 17.89541 41.59028
## 5    t13 17.94329 41.81583
## 6    t14 17.95515 40.93829
## 7    t15 18.49101 40.78501
## 8    t16 18.50241 40.74782
## 9    t17 17.57091 41.08303
## 10   t18 18.25691 41.74148
## 11   t19 18.94612 40.05859
## 12    t2 18.71446 40.87573
## 13   t20 16.76495 37.15394
## 14   t21 18.90497 40.41419
## 15   t22 17.34779 37.23577
## 16   t23 17.94877 41.79086
## 17   t24 18.50319 40.05271
## 18   t25 17.81249 39.34886
## 19    t3 18.49510 39.82282
## 20    t4 18.23416 41.28766
## 21    t5 16.85516 36.93270
## 22    t6 18.14229 40.01092
## 23    t7 17.68522 39.74654
## 24    t8 18.54884 40.72576
## 25    t9 18.10201 41.73898
```

Check if the `tip.label` in the `phylo` object have the same row order as `traits$Taxon`

```
cbind(traits$Taxon, tree$tip.label)
```

```
##       [,1]  [,2] 
##  [1,] "t1"  "t25"
##  [2,] "t10" "t6" 
##  [3,] "t11" "t7" 
##  [4,] "t12" "t3" 
##  [5,] "t13" "t8" 
##  [6,] "t14" "t12"
##  [7,] "t15" "t1" 
##  [8,] "t16" "t15"
##  [9,] "t17" "t16"
## [10,] "t18" "t21"
## [11,] "t19" "t2" 
## [12,] "t2"  "t11"
## [13,] "t20" "t4" 
## [14,] "t21" "t13"
## [15,] "t22" "t18"
## [16,] "t23" "t17"
## [17,] "t24" "t9" 
## [18,] "t25" "t10"
## [19,] "t3"  "t23"
## [20,] "t4"  "t14"
## [21,] "t5"  "t24"
## [22,] "t6"  "t19"
## [23,] "t7"  "t22"
## [24,] "t8"  "t5" 
## [25,] "t9"  "t20"
```

The species data in the `traits` data.frame must be in the same row order as the names of the species in the tree object (species names are in `tree$tip.label`). To accompish this, begin by changing the `rownames()` in the `traits` data.frame to the tip labels

```r
rownames(traits) <- traits$Taxon
```

Run the following command to ensure that everything matches

```r
traits <- traits[match(tree$tip.label, rownames(traits)), ]
traits
```

```
##     Taxon   trait1   trait2
## t25   t25 17.81249 39.34886
## t6     t6 18.14229 40.01092
## t7     t7 17.68522 39.74654
## t3     t3 18.49510 39.82282
## t8     t8 18.54884 40.72576
## t12   t12 17.89541 41.59028
## t1     t1 18.32059 40.62082
## t15   t15 18.49101 40.78501
## t16   t16 18.50241 40.74782
## t21   t21 18.90497 40.41419
## t2     t2 18.71446 40.87573
## t11   t11 18.45661 40.78300
## t4     t4 18.23416 41.28766
## t13   t13 17.94329 41.81583
## t18   t18 18.25691 41.74148
## t17   t17 17.57091 41.08303
## t9     t9 18.10201 41.73898
## t10   t10 17.58344 41.70659
## t23   t23 17.94877 41.79086
## t14   t14 17.95515 40.93829
## t24   t24 18.50319 40.05271
## t19   t19 18.94612 40.05859
## t22   t22 17.34779 37.23577
## t5     t5 16.85516 36.93270
## t20   t20 16.76495 37.15394
```

Check again to make sure that `tip.label` in the tree phylo object have same row order as `traits$Taxon`

```r
cbind(traits$Taxon, tree$tip.label)
```

```
##       [,1]  [,2] 
##  [1,] "t25" "t25"
##  [2,] "t6"  "t6" 
##  [3,] "t7"  "t7" 
##  [4,] "t3"  "t3" 
##  [5,] "t8"  "t8" 
##  [6,] "t12" "t12"
##  [7,] "t1"  "t1" 
##  [8,] "t15" "t15"
##  [9,] "t16" "t16"
## [10,] "t21" "t21"
## [11,] "t2"  "t2" 
## [12,] "t11" "t11"
## [13,] "t4"  "t4" 
## [14,] "t13" "t13"
## [15,] "t18" "t18"
## [16,] "t17" "t17"
## [17,] "t9"  "t9" 
## [18,] "t10" "t10"
## [19,] "t23" "t23"
## [20,] "t14" "t14"
## [21,] "t24" "t24"
## [22,] "t19" "t19"
## [23,] "t22" "t22"
## [24,] "t5"  "t5"
## [25,] "t20" "t20"
```

Okay now we're ready for analysis!

Naive analysis of trait data

Extract trait data from `traits` data.frame

```
trait1 <- traits$trait1
trait1
```

```
##  [1] 17.81249 18.14229 17.68522 18.49510 18.54884 17.89541 18.32059
##  [8] 18.49101 18.50241 18.90497 18.71446 18.45661 18.23416 17.94329
## [15] 18.25691 17.57091 18.10201 17.58344 17.94877 17.95515 18.50319
## [22] 18.94612 17.34779 16.85516 16.76495
```

```
trait2 <- traits$trait2
trait2
```

```
##  [1] 39.34886 40.01092 39.74654 39.82282 40.72576 41.59028 40.62082
##  [8] 40.78501 40.74782 40.41419 40.87573 40.78300 41.28766 41.81583
## [15] 41.74148 41.08303 41.73898 41.70659 41.79086 40.93829 40.05271
## [22] 40.05859 37.23577 36.93270 37.15394
```

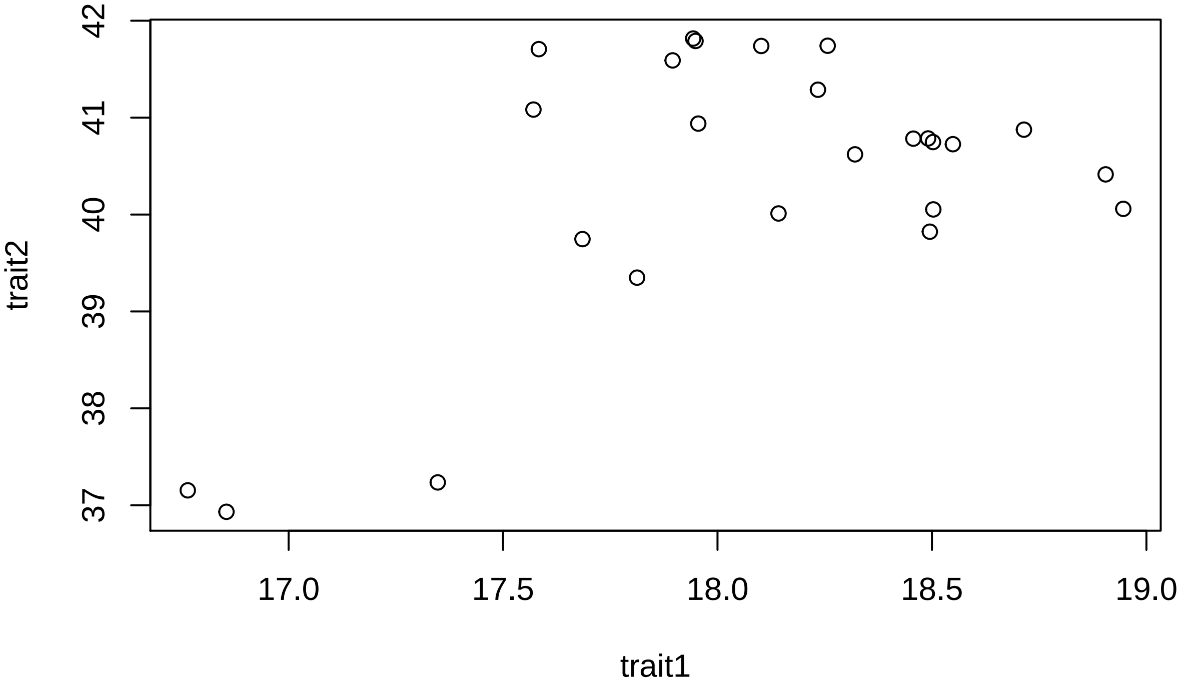
FYI: sometimes we need to compute relative traits (e.g., brain size adjusted for body size). This is easily done by dividing columns

```
relative_trait1 <- trait1/trait2
relative_trait1
```

```
##  [1] 0.4526812 0.4534335 0.4449499 0.4644348 0.4554573 0.4302788 0.4510148
##  [8] 0.4533777 0.4540711 0.4677805 0.4578379 0.4525565 0.4416371 0.4291027
## [15] 0.4373805 0.4276926 0.4336956 0.4215987 0.4294902 0.4385907 0.4619711
## [22] 0.4729603 0.4658906 0.4563748 0.4512293
```

Plot the trait data to visualize the data

```
plot(trait1, trait2)
```



Transform the trait data with natural log. Plot the ln(data)

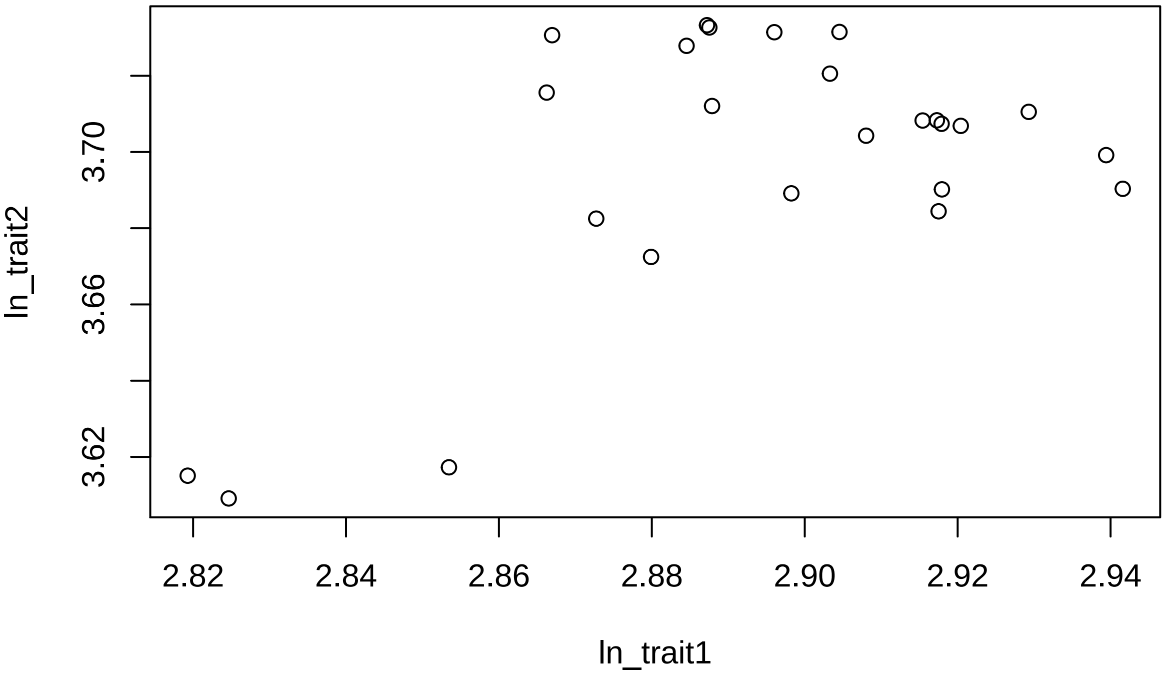
```
ln_trait1 <- log(trait1)
ln_trait1
```

```
##  [1] 2.879900 2.898246 2.872729 2.917506 2.920407 2.884544 2.908026
##  [8] 2.917285 2.917901 2.939425 2.929296 2.915423 2.903297 2.887216
## [15] 2.904544 2.866245 2.896023 2.866958 2.887521 2.887877 2.917943
## [22] 2.941599 2.853465 2.824657 2.819290
```

```
ln_trait2 <- log(trait2)
ln_trait2
```

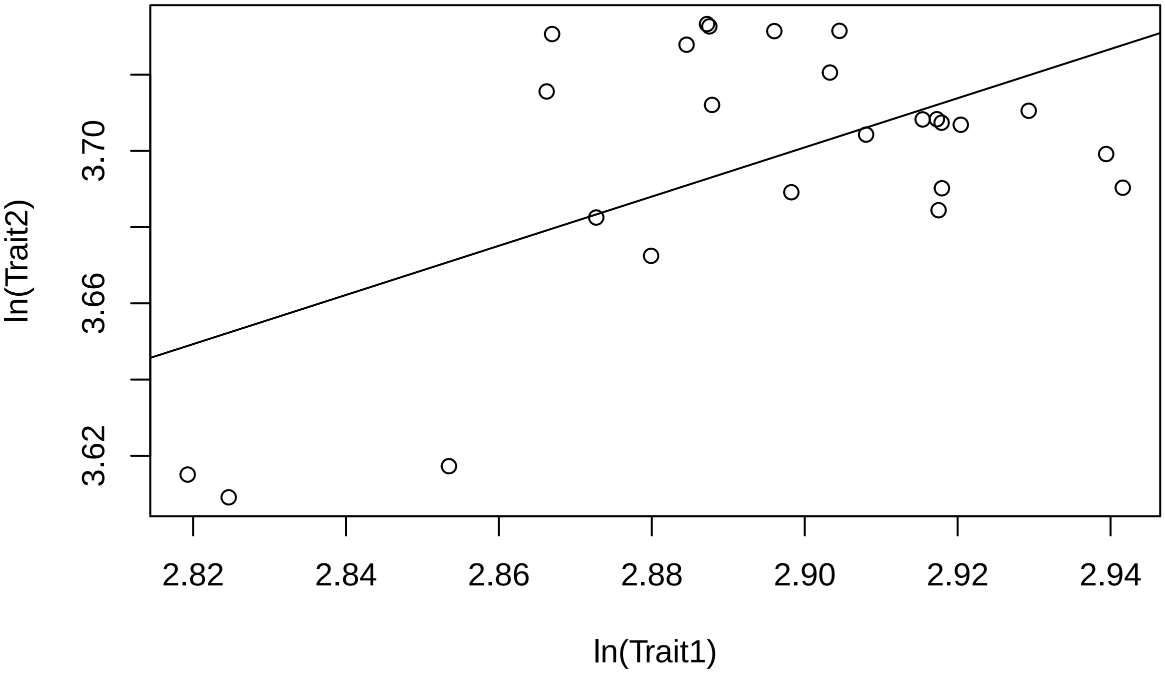
```
##  [1] 3.672467 3.689153 3.682523 3.684440 3.706861 3.727866 3.704281
##  [8] 3.708315 3.707402 3.699181 3.710536 3.708265 3.720564 3.733275
## [15] 3.731495 3.715595 3.731436 3.730659 3.732678 3.712066 3.690196
## [22] 3.690343 3.617270 3.609097 3.615070
```

```
plot(ln_trait1, ln_trait2)
```



Fit a linear model to the ln(data)

```
fit <- lm(log(trait2) ~ log(trait1))
plot(ln_trait1, ln_trait2, xlab = "ln(Trait1)", ylab = "ln(Trait2)")
abline(fit)
```



Question: What is the relationship between trait 1 and trait 2?

```
summary(fit)
```

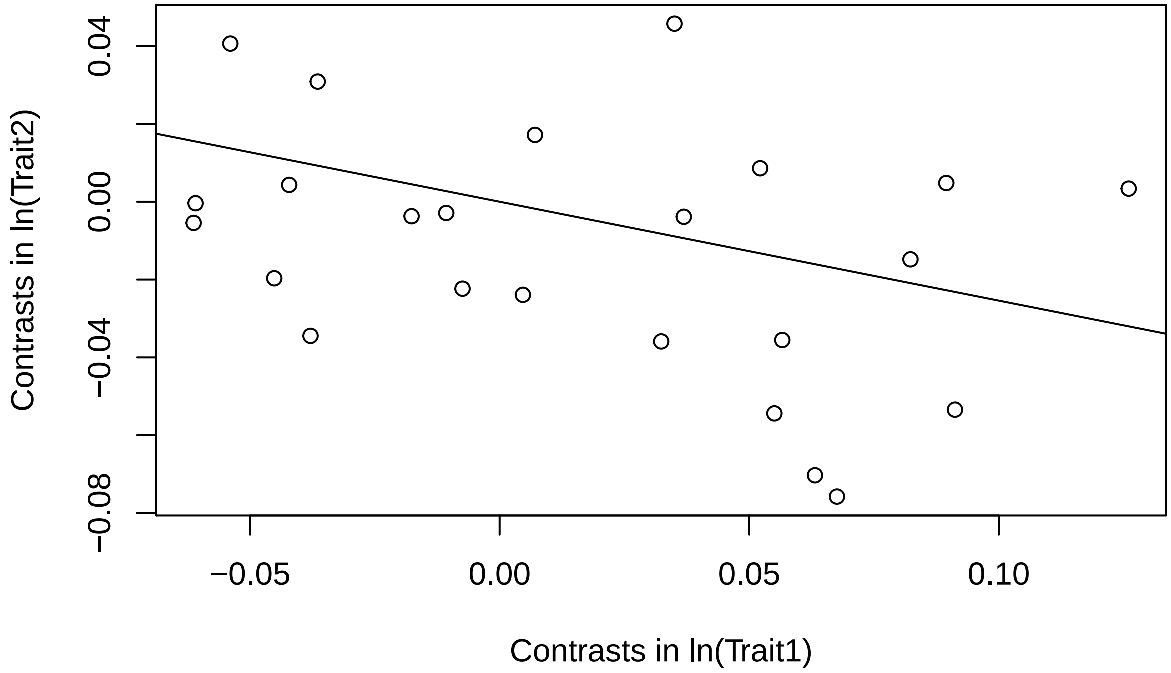
```
##
## Call:
```

```
## lm(formula = log(trait2) ~ log(trait1))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.053624 -0.022307 -0.003764  0.027639  0.051058
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.8295     0.5656   3.234  0.00366 **
## log(trait1)   0.6453     0.1954   3.302  0.00311 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03014 on 23 degrees of freedom
## Multiple R-squared:  0.3217, Adjusted R-squared:  0.2922
## F-statistic: 10.91 on 1 and 23 DF,  p-value: 0.003112
```

Question: What can we conclude from the naive analysis about the relationship between trait 1 and trait 2?

Analysis with PICs

```
PIC_ln_trait1 <- pic(ln_trait1, tree)
PIC_ln_trait2 <- pic(ln_trait2, tree)
PIC_fit <- lm(PIC_ln_trait2 ~ PIC_ln_trait1 - 1) # "-1 " removes intercept term
plot(PIC_ln_trait1, PIC_ln_trait2,
  xlab = "Contrasts in ln(Trait1)",
  ylab = "Contrasts in ln(Trait2)")
abline(PIC_fit)
```



Question: What can we conclude about the relationship between the PICs for trait 1 and trait 2?

```
summary(PIC_fit)
```

```
##
```

```
## Call:
## lm(formula = PIC_ln_trait2 ~ PIC_ln_trait1 - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.05859 -0.02832 -0.01203  0.01963  0.05466
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## PIC_ln_trait1  -0.2541     0.1112  -2.285   0.0318 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03099 on 23 degrees of freedom
## Multiple R-squared:  0.1851, Adjusted R-squared:  0.1496
## F-statistic: 5.223 on 1 and 23 DF,  p-value: 0.03184
```

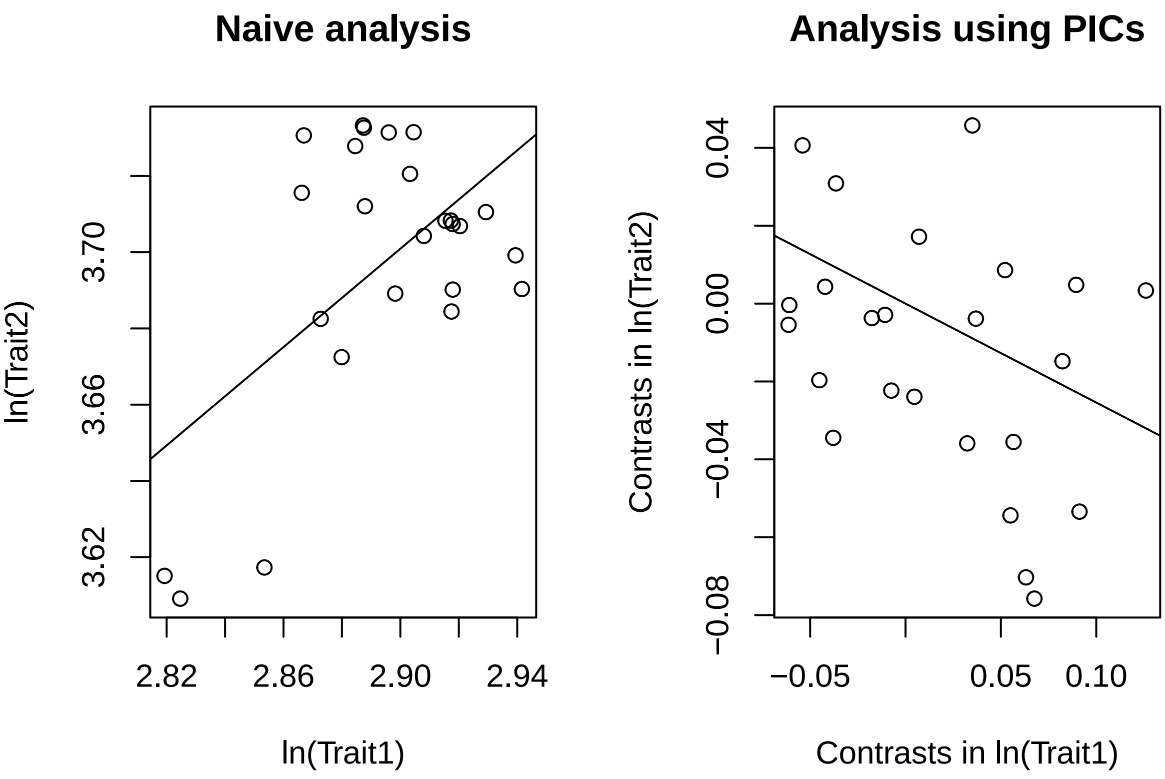
Question: Why do we fit the linear model without an intercept term?

Prepare plots for a report

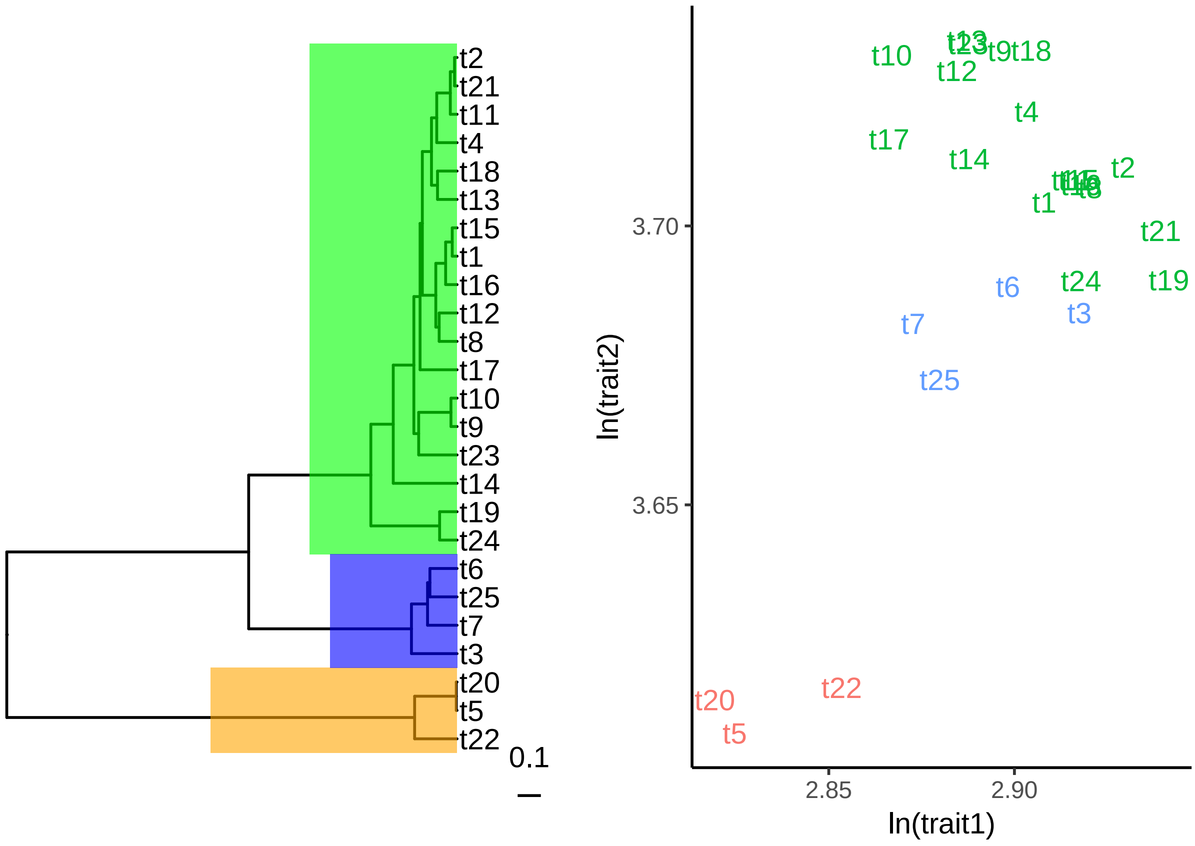
```
par(mfrow = c(1, 2)) #This command allows you to plot two figures side-by-side.

# Plotting naive analysis
plot(ln_trait1, ln_trait2,
  xlab = "ln(Trait1)", ylab = "ln(Trait2)",
  main = "Naive analysis")
abline(fit)

# Plotting Analysis using PICs
plot(PIC_ln_trait1, PIC_ln_trait2,
  xlab = "Contrasts in ln(Trait1)", ylab = "Contrasts in ln(Trait2)",
  main = "Analysis using PICs")
abline(PIC_fit)
```



Understanding the intuition



Some Fancy stuff you can do

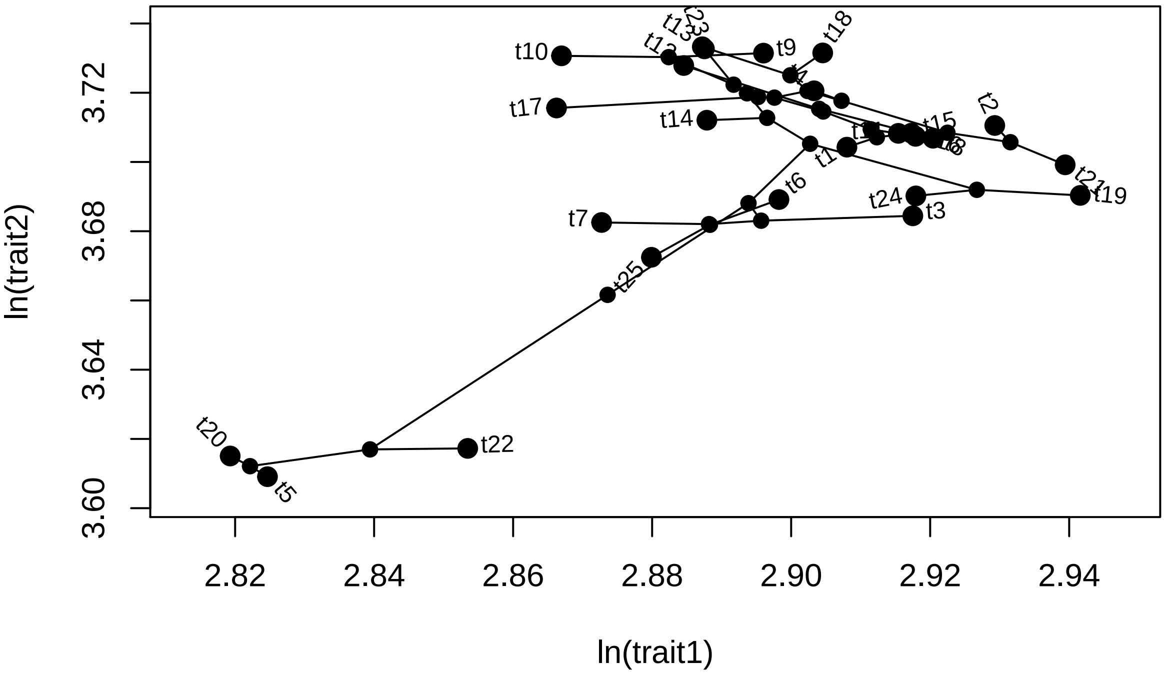
```
library(phytools)
```

```
## Loading required package: maps

##
## Attaching package: 'phytools'

## The following objects are masked from 'package:ggtree':
##
##     read.newick, reroot
```

```
ln_traits = cbind(ln_trait1, ln_trait2)
rownames(ln_traits) <- traits$Taxon
phylomorphospace(tree = tree, ln_traits, xlab = "ln(trait1)", ylab = "ln(trait2)")
```



Homework

Courtesy of Tyler McCraney

Why do some bats have larger testes than others?

Reanalyze the trait data *batTraits.csv* from Hosken (1998) using the updated bat tree *batTree* by Shi and Rabosky (2015).

Use relative testes mass for your analysis (*i.e.*, testes mass / body mass)

Conduct both a (1) naive analysis, and (2) proper analyses using PICs

Report the p-value and adjusted R-squared of each analysis in the text of your lab report