

Sperm size figures

jeremy

Wednesday, January 14, 2015

Sperm size figures

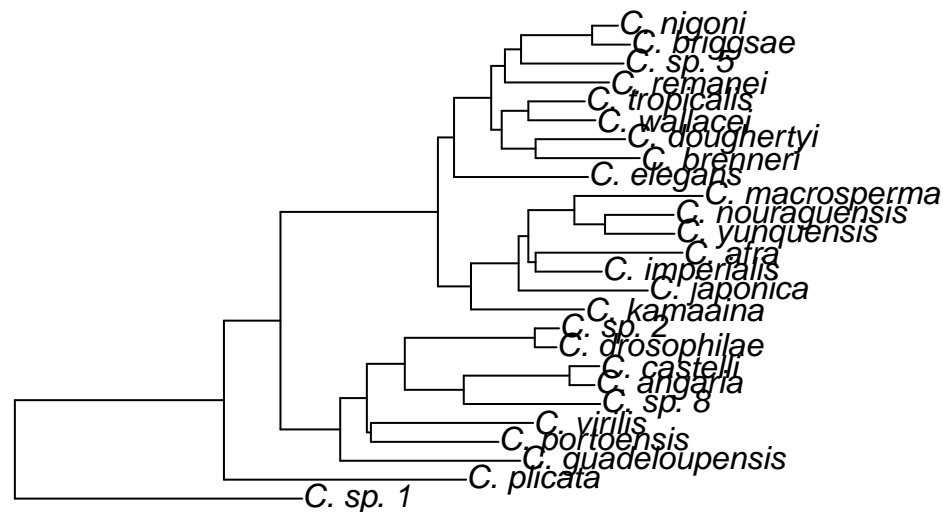
This is the output of an R markdown script - a fancy way of embedding r in a page so we can rerun/tune it as needed and then have the commands

I'm embedding, but hiding a bunch of stuff for readability - the actual code is in the .Rmd file.

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
## Warning: package 'survival' was built under R version 3.1.3
```

Here's the tree from Kiontke et al 2011, figure 3 initial values I recieved had angaria and sp12 flipped - I've corrected it here Here's the tree:



Now read in the compiled data - we have oocytesize, spermsize, primaryspermatocytesize, and oocytearea. The species names in the files are inconsistent - they are fixed now. For now, the oocyte area is using the area formulation as made by the initial sperm size formula. All response variables are logged unless otherwise noted.

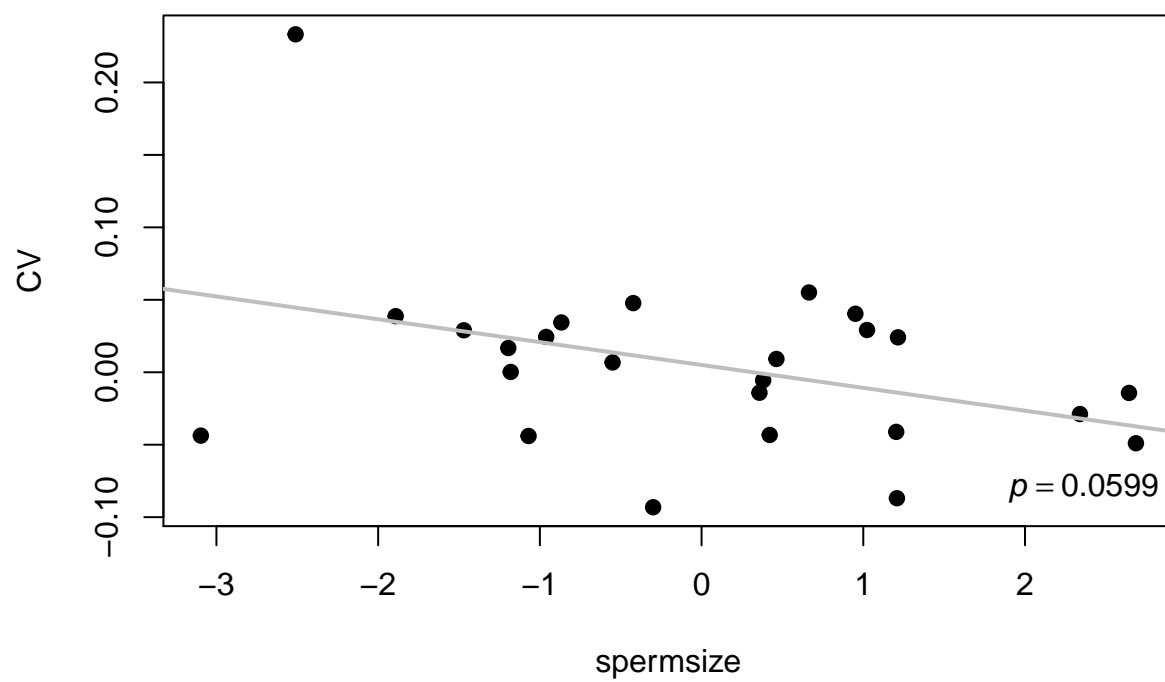
I have filtered out some of the data: Any oocytesize that is not stage -1, and any primaryspermatocyte data that is from hermaphrodites rather than males.

We now have clean data, clustered into species, strains or individuals. Variables ending in 1 are in species, 2 are strains and 3 are individuals. The non-clustered data is as is.

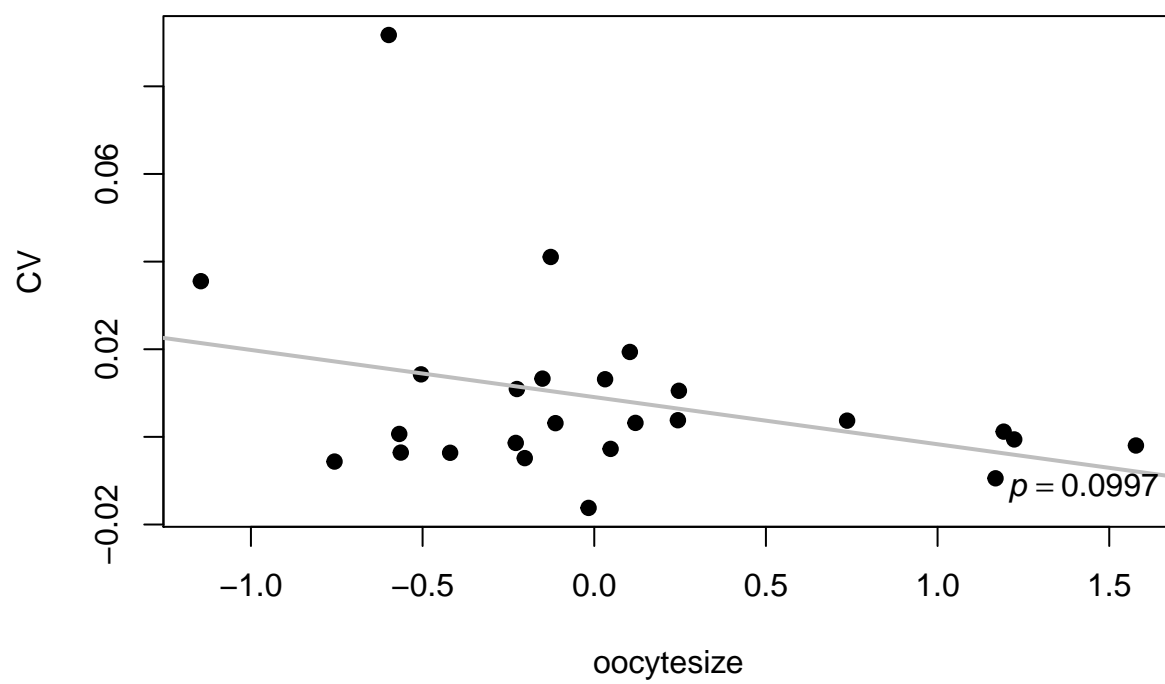
PIC plots - mean vs CV

Plotting the phylogenetic independent contrasts (PIC) of means of traits vs CV of traits. All means and CVs are grouped at strain level.

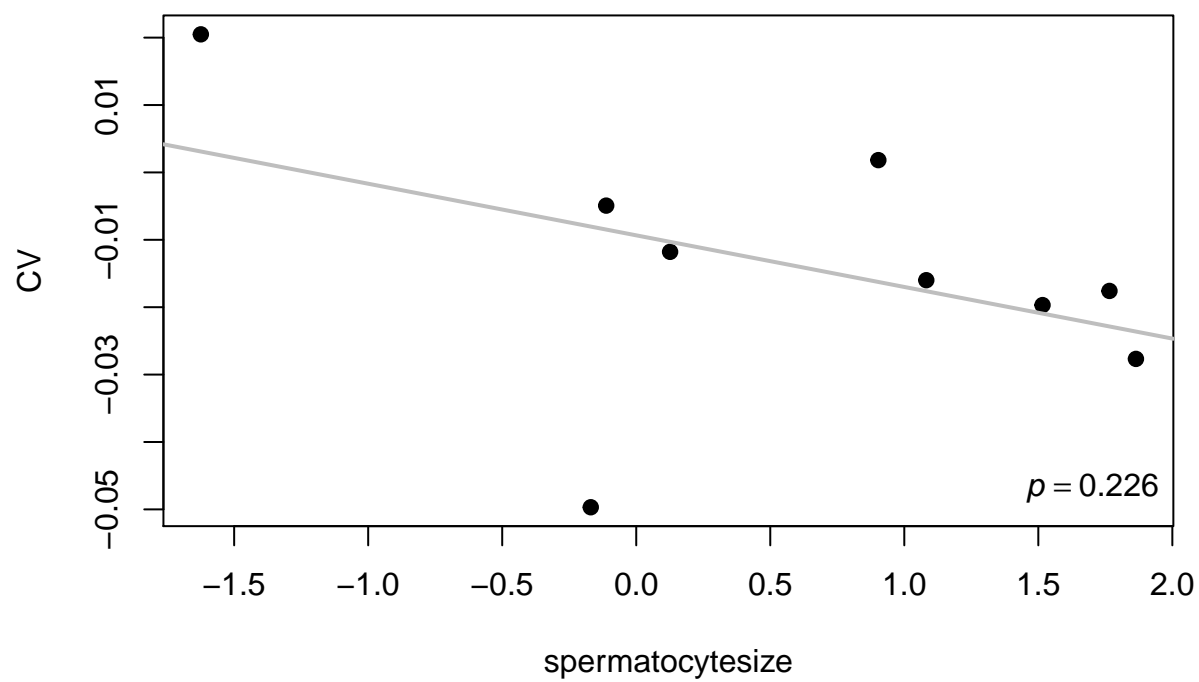
PIC of spermsize means vs CV



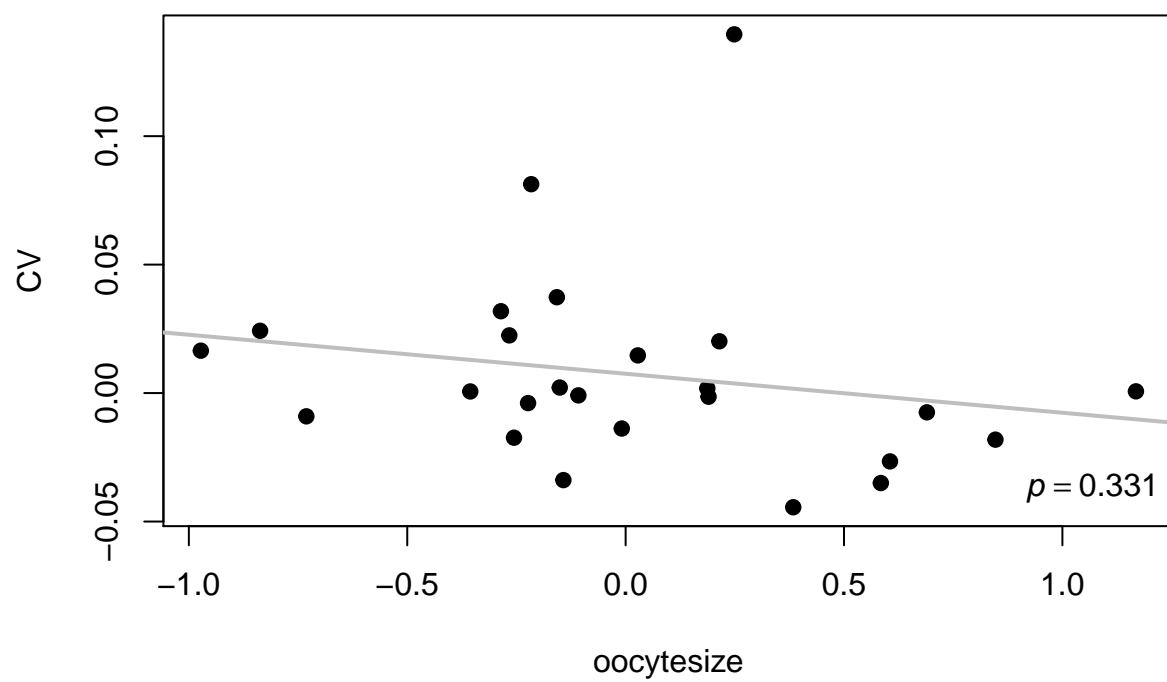
PIC of oocytesize means vs CV



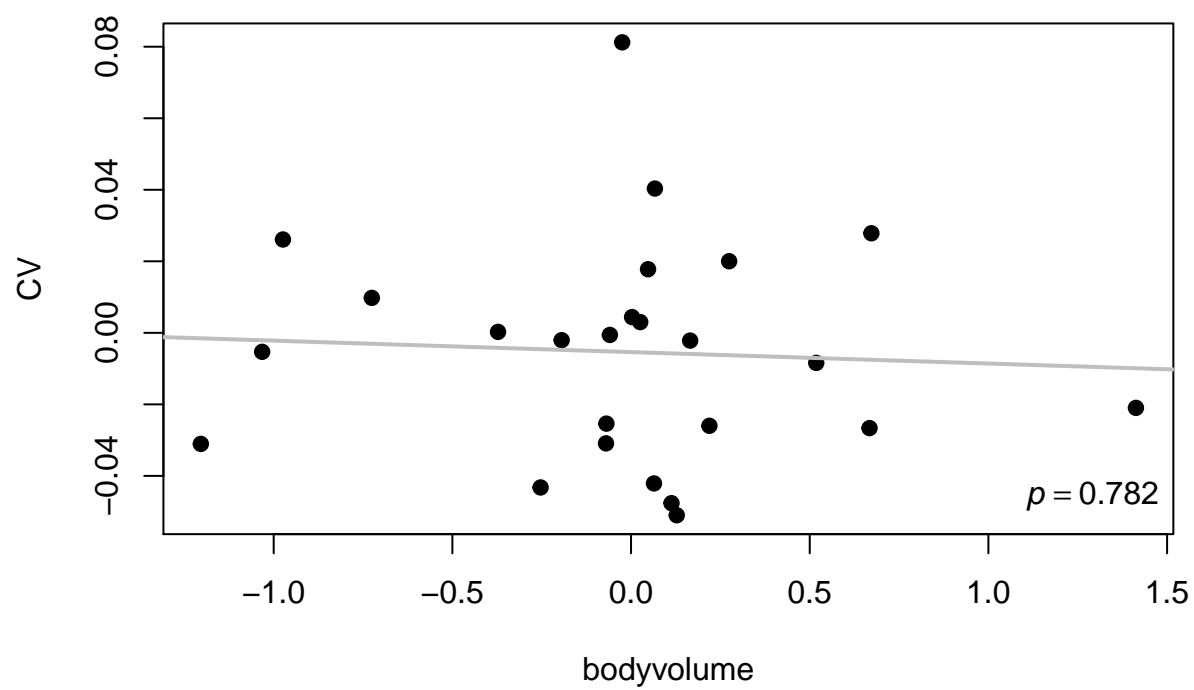
PIC of spermatocytesize means vs CV



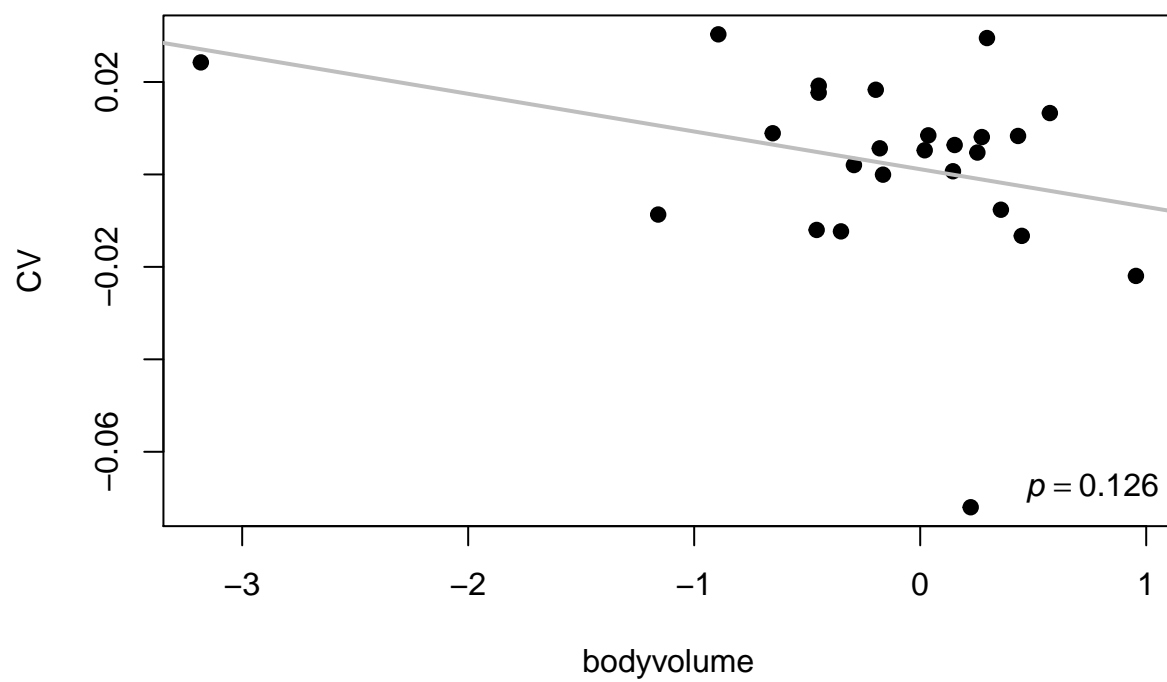
PIC of oocytesizearea means vs CV



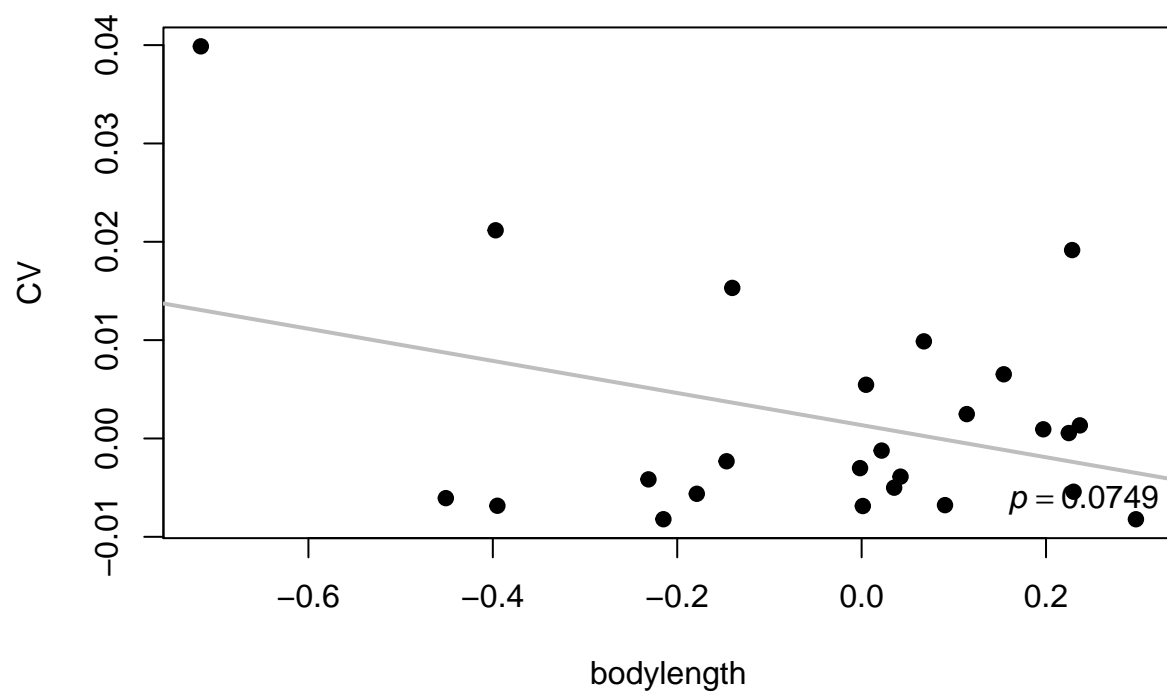
PIC cv bodyvolume male



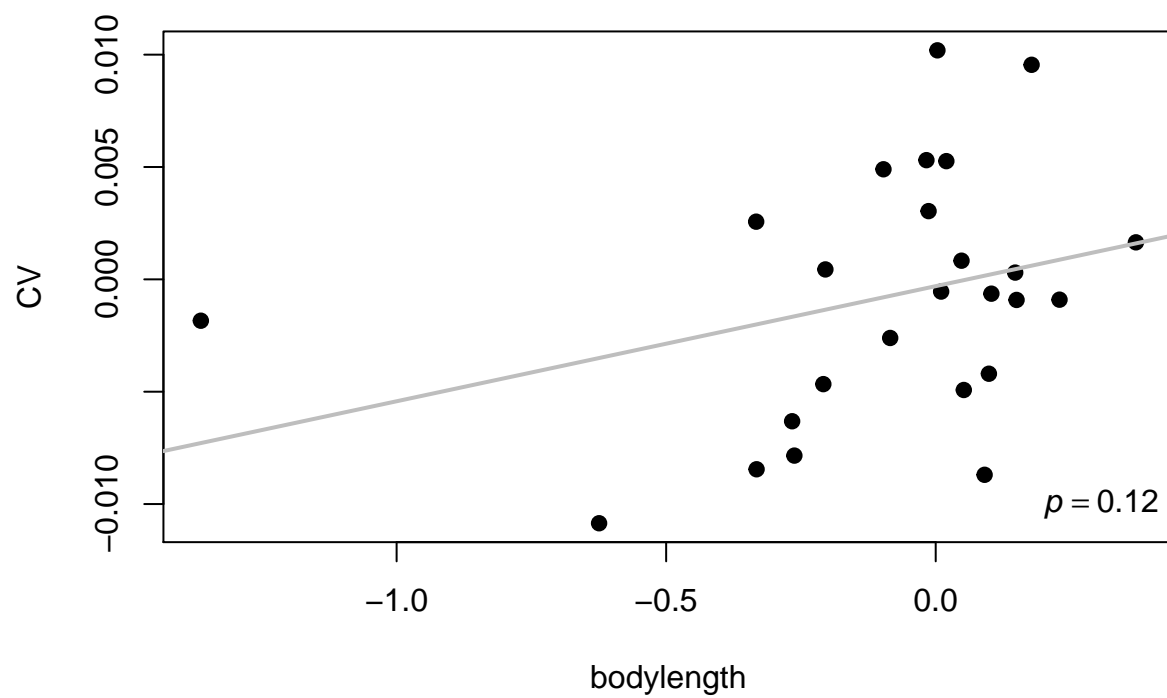
PIC cv bodyvolume female



PIC cv bodylength male

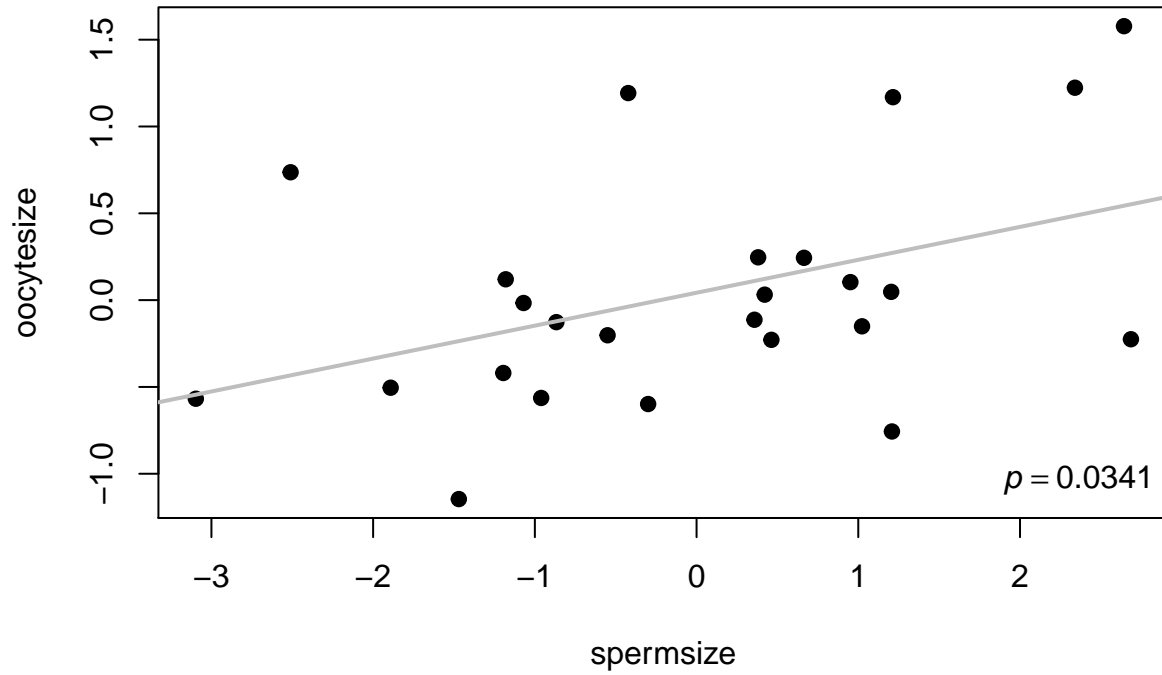


PIC cv bodylength female

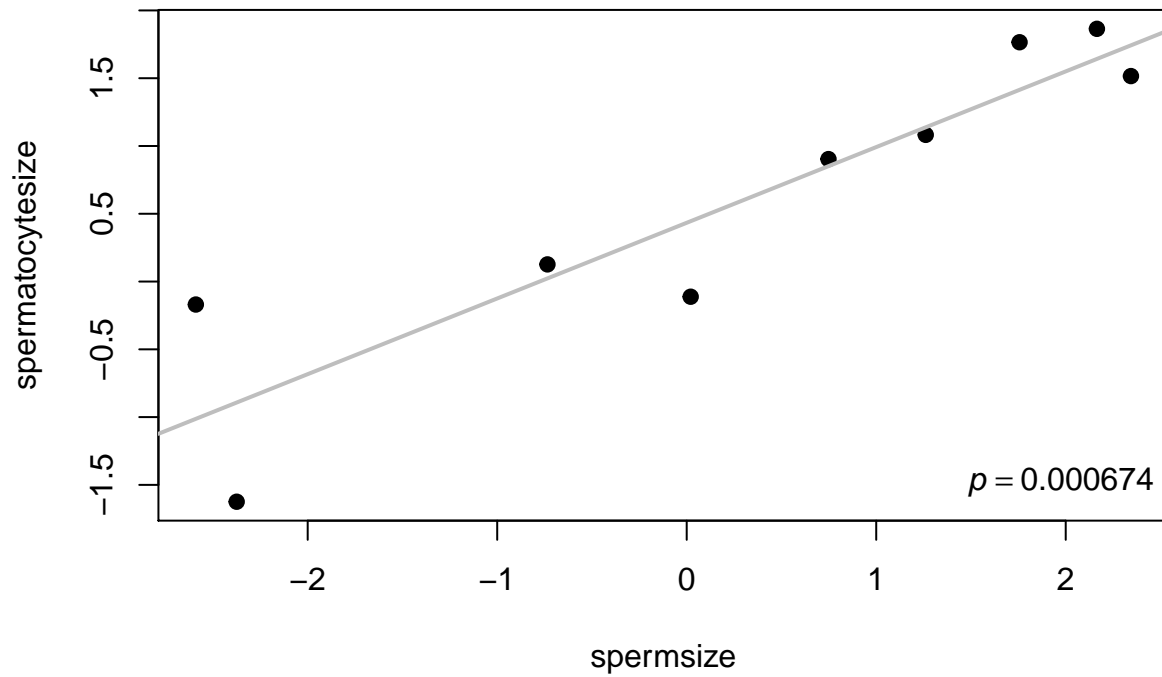


PIC plots - traits vs traits

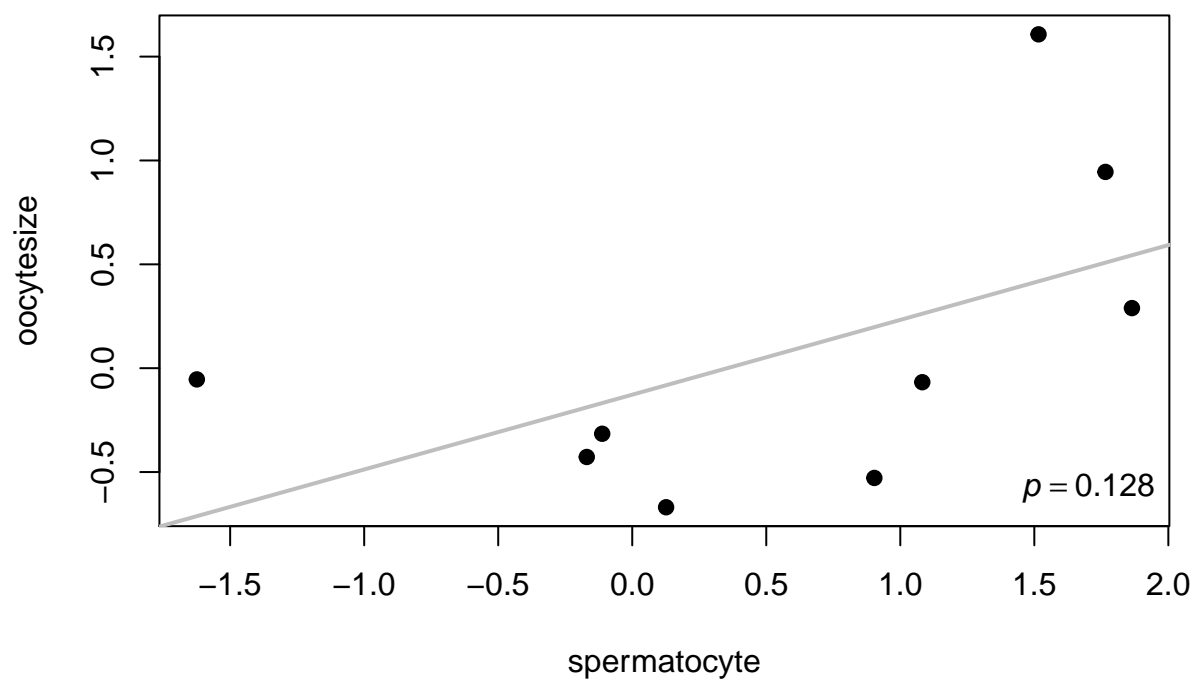
PIC of spermsize vs oocytesize



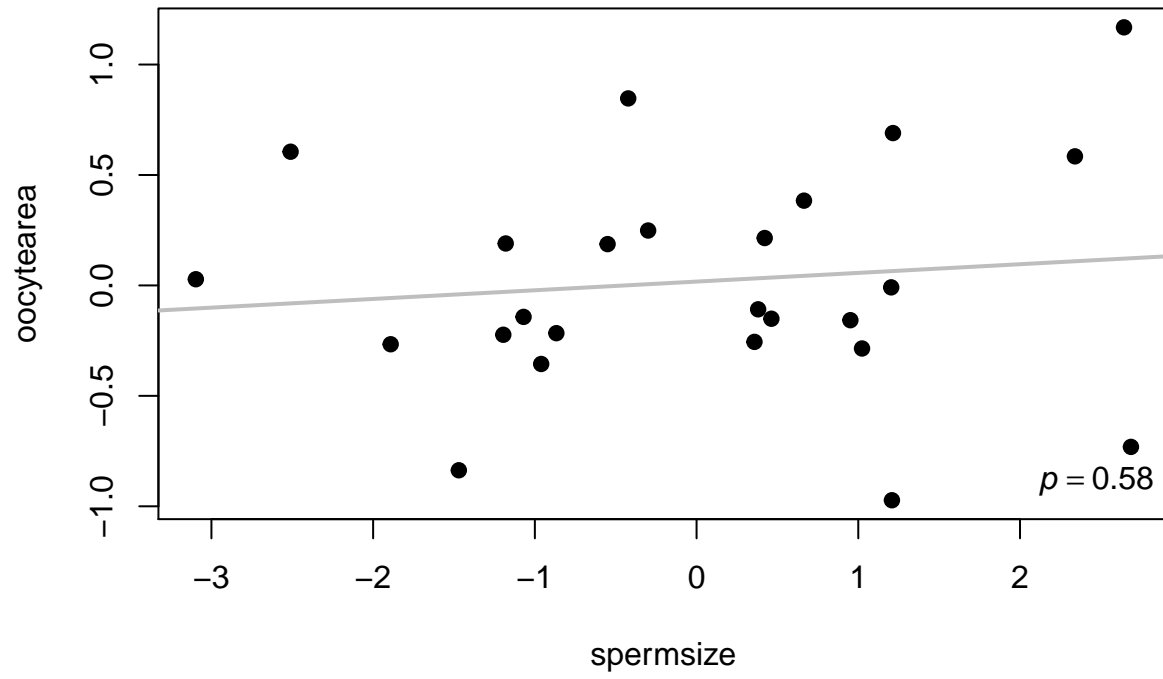
PIC of spermsize vs spermatocytesize



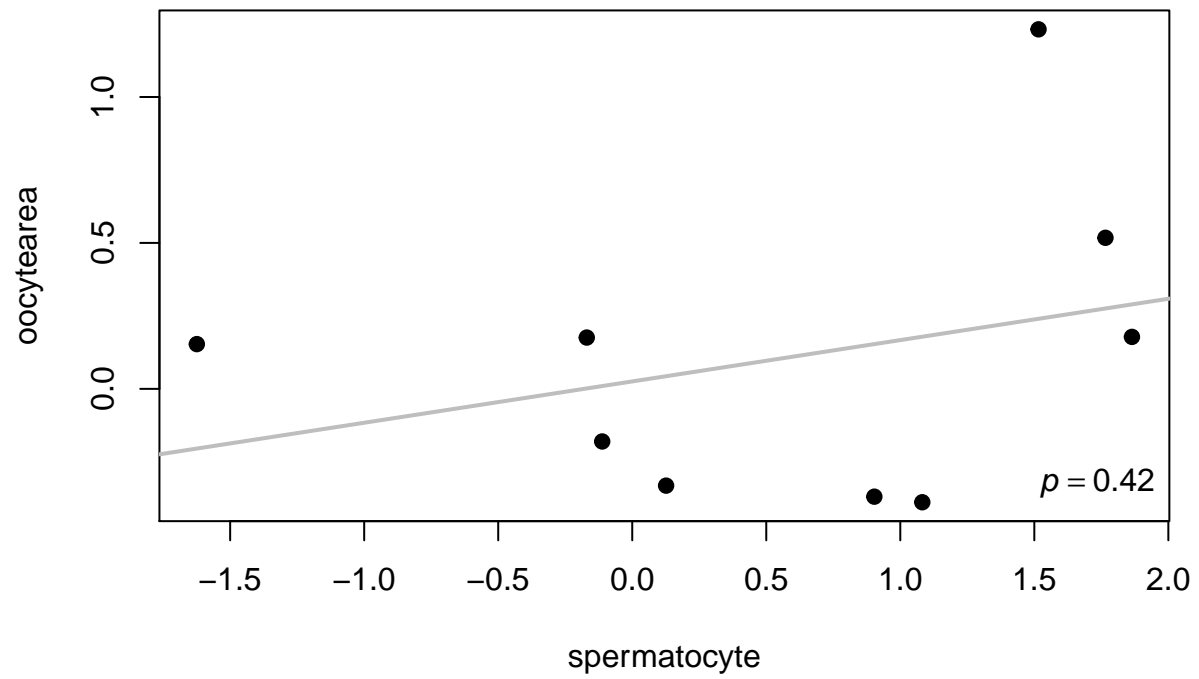
PIC of spermatocytesize vs oocytesize



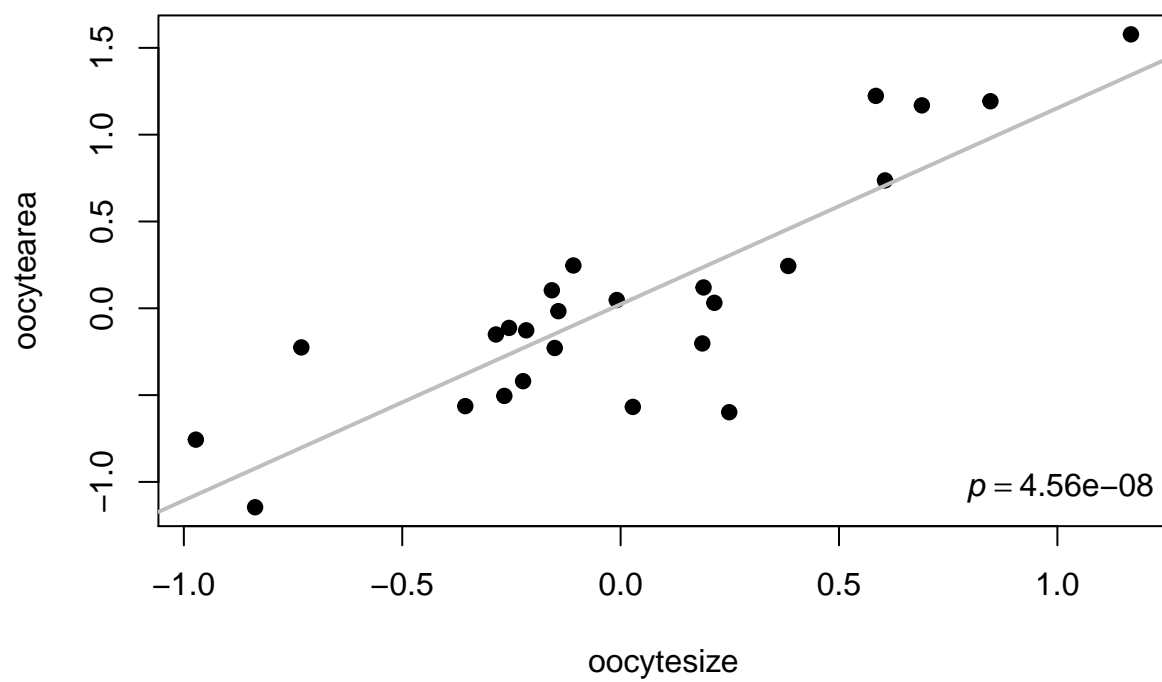
PIC of spermsize vs oocytearea



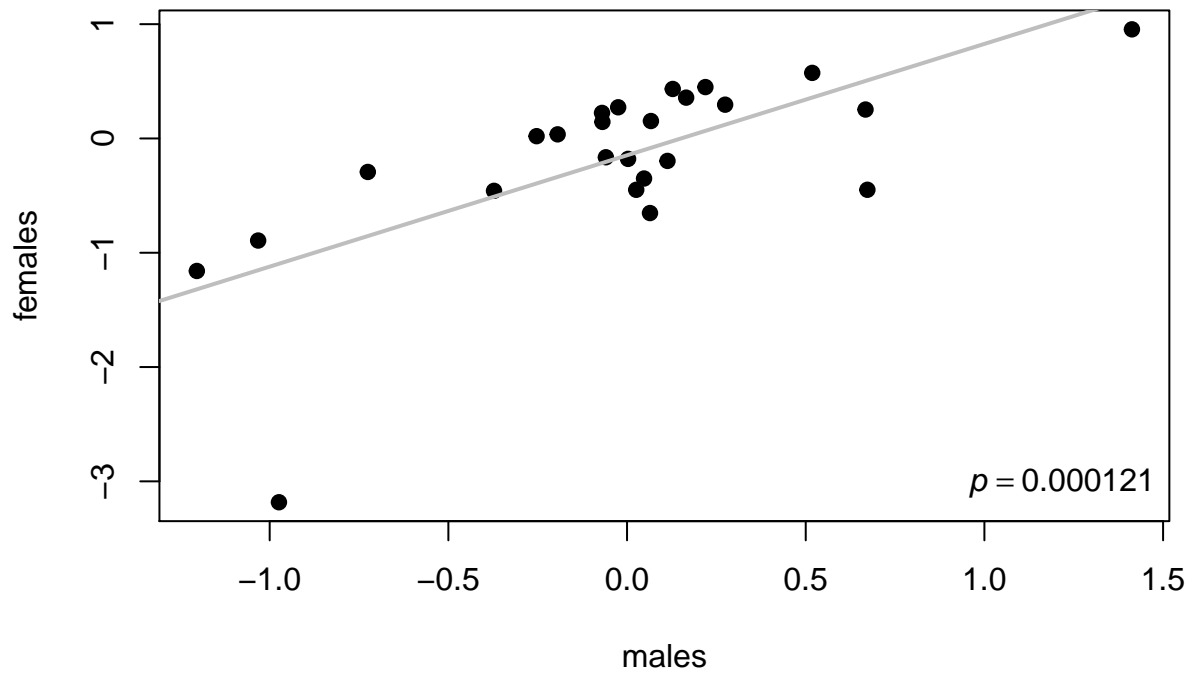
PIC of spermatocytesize vs oocytearea



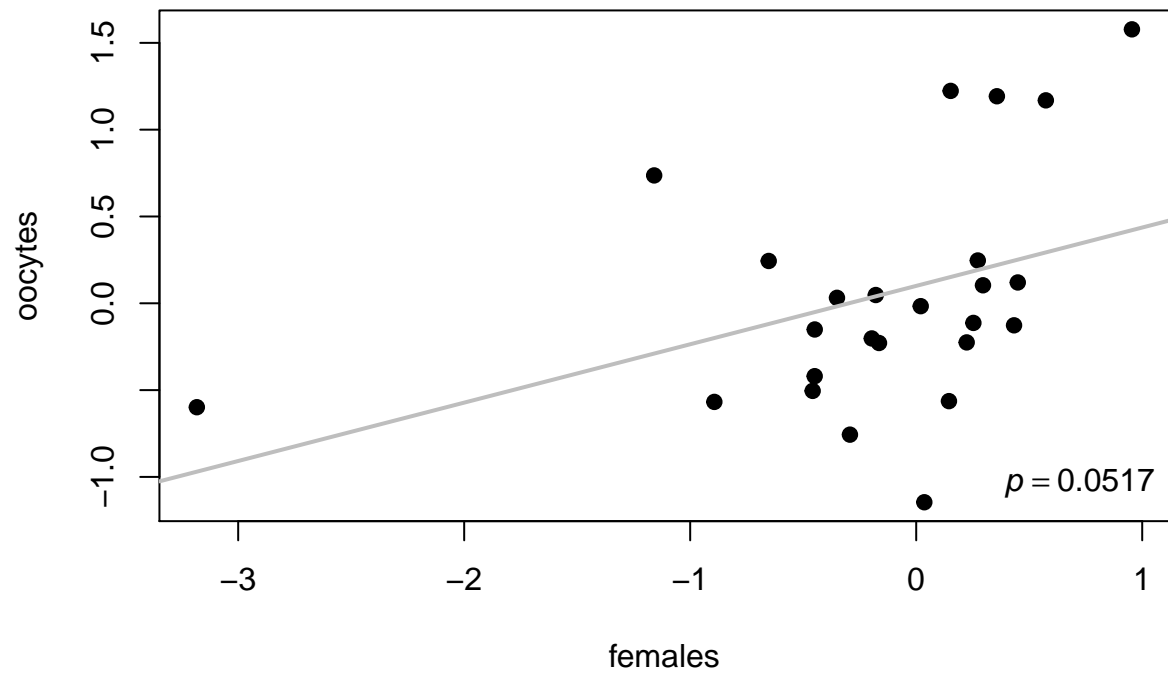
PIC of oocytesize vs oocytearea



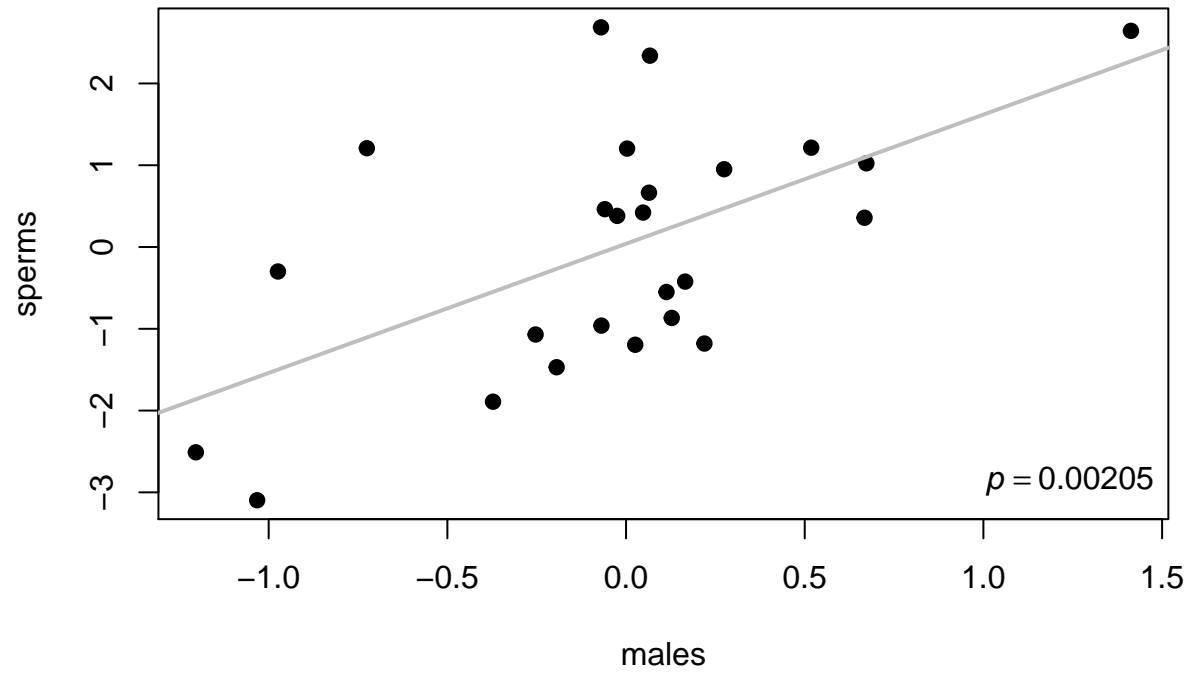
Body area male vs female



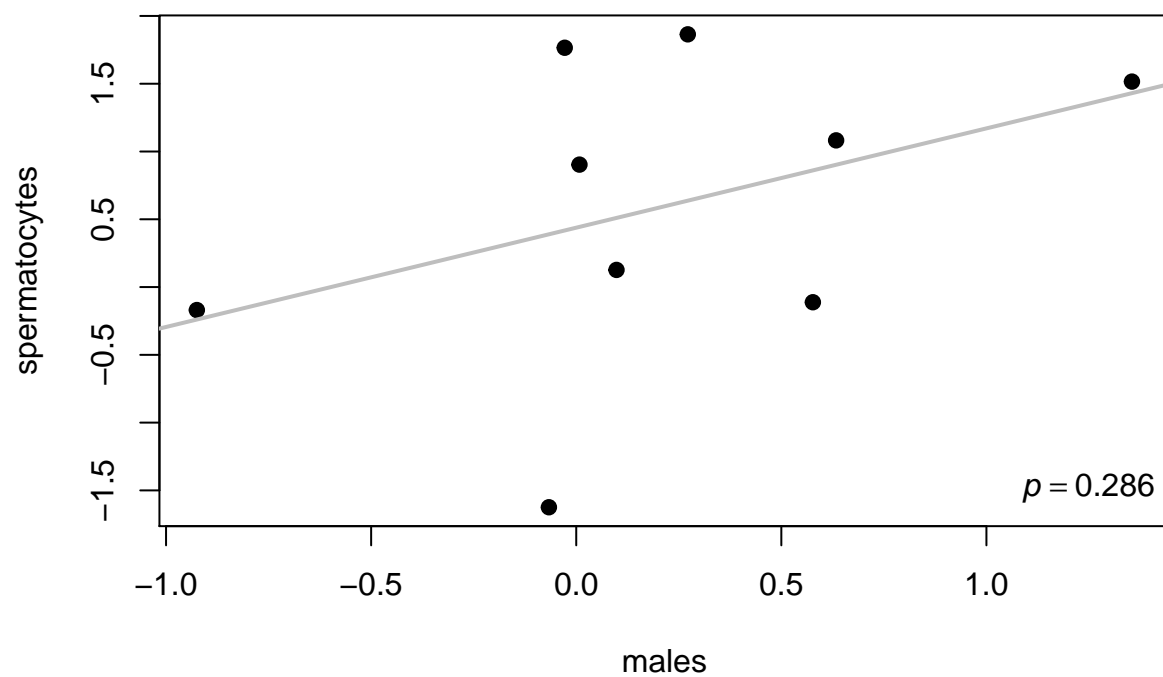
Body area female vs oocytesize



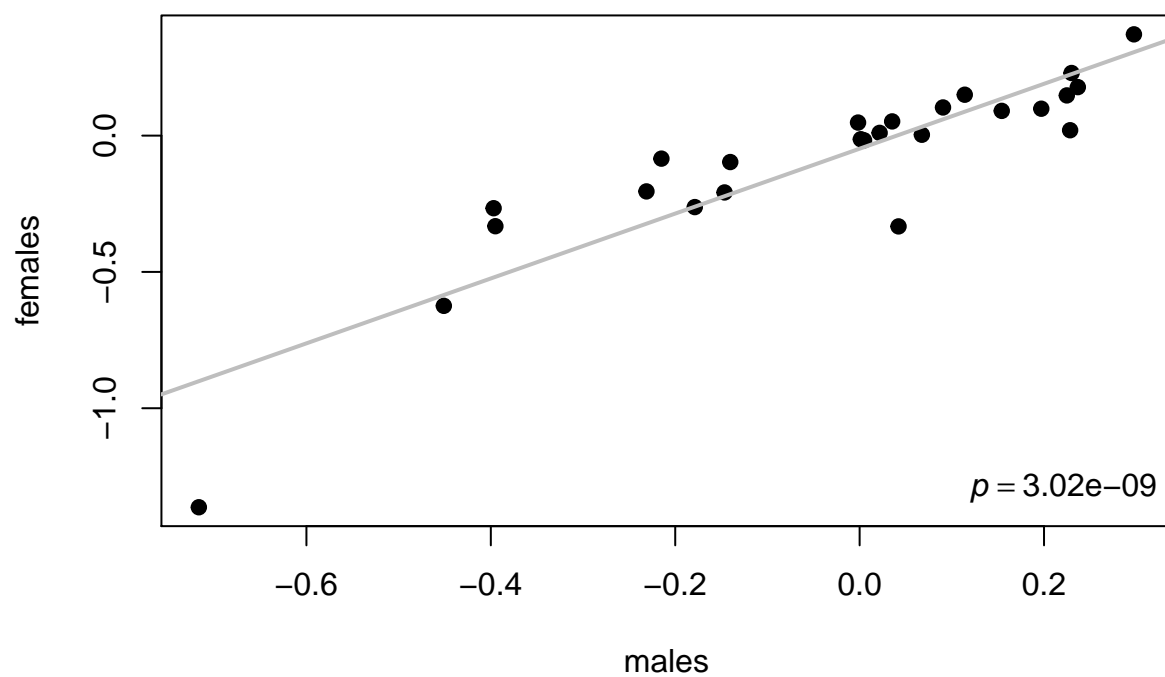
Body area male vs spermsize



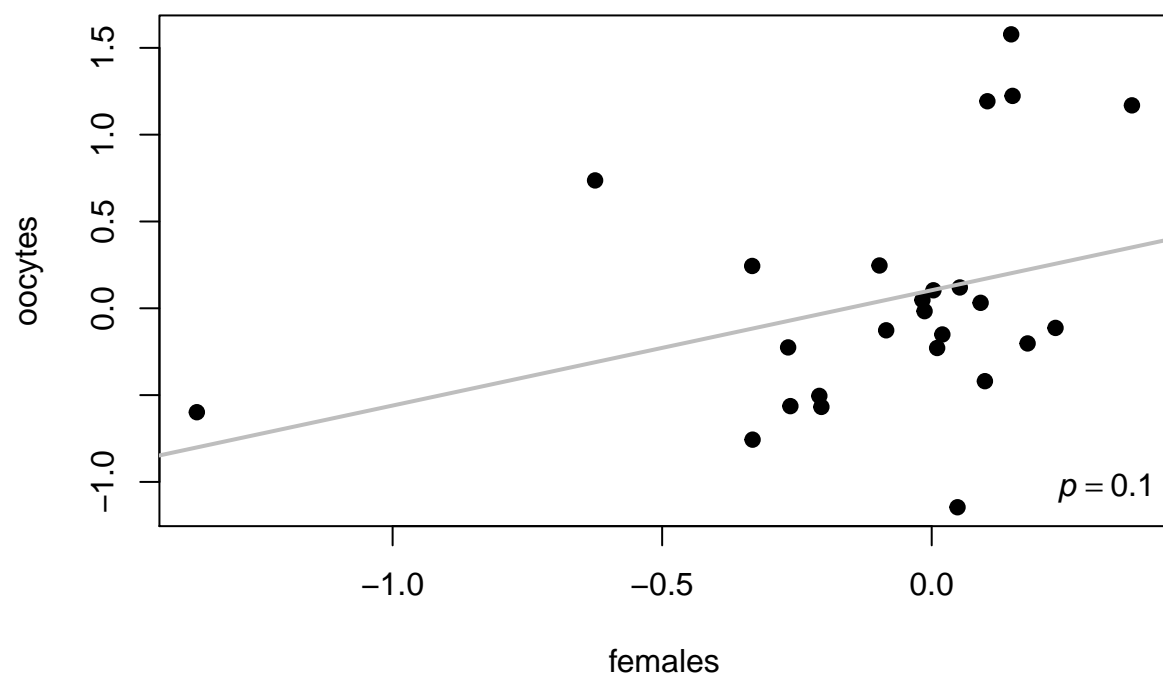
Body area male vs spermatocytesize



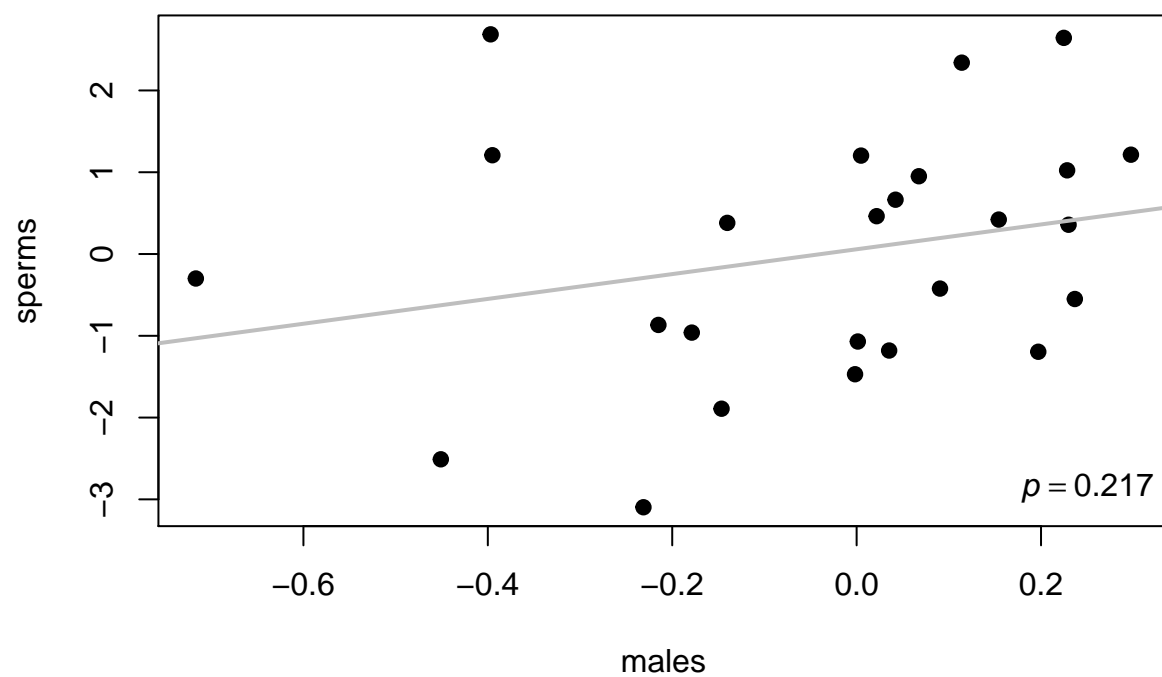
Body length male vs female



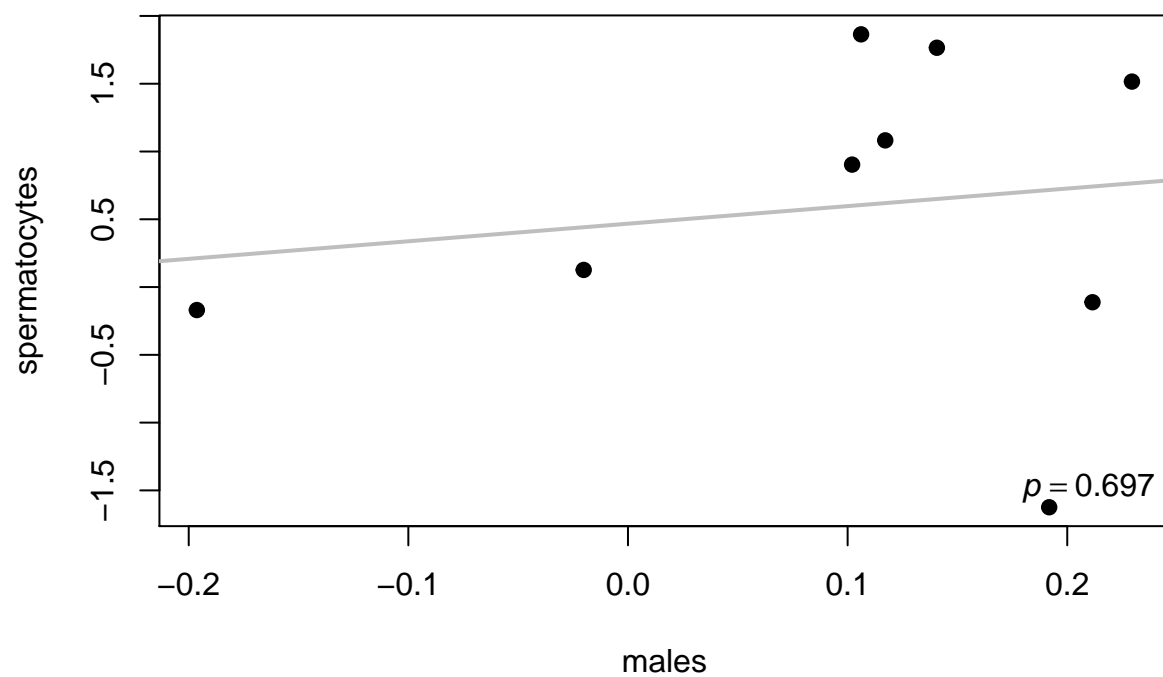
Body length female vs oocytesize



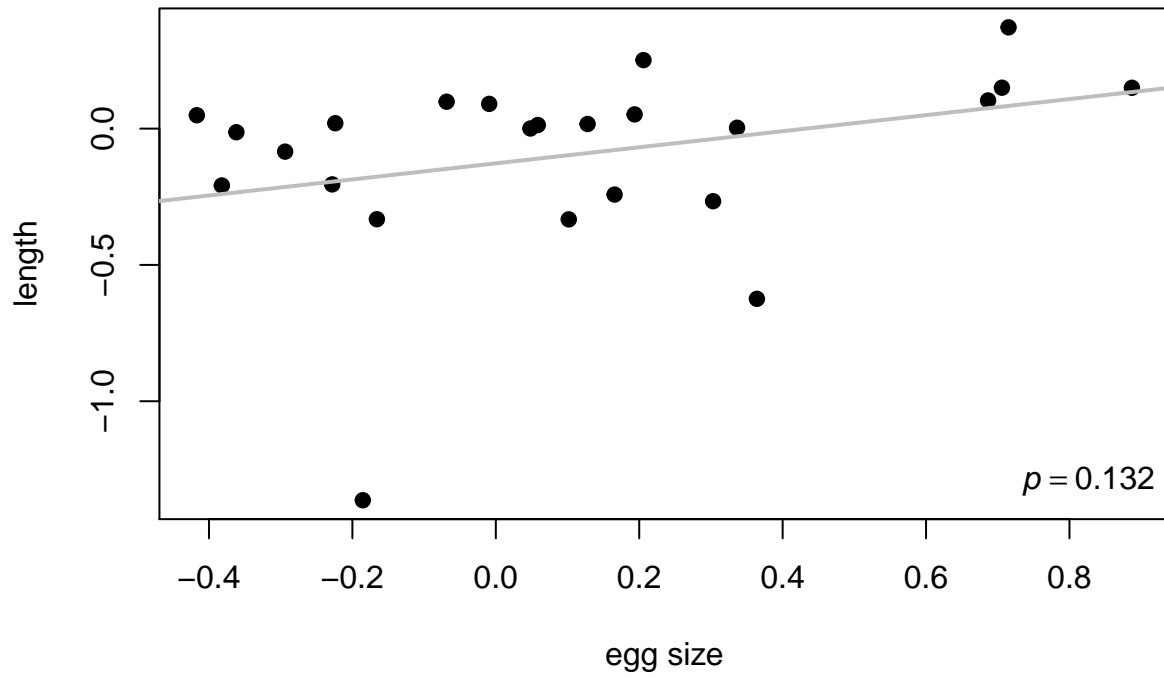
Body length male vs spermsize



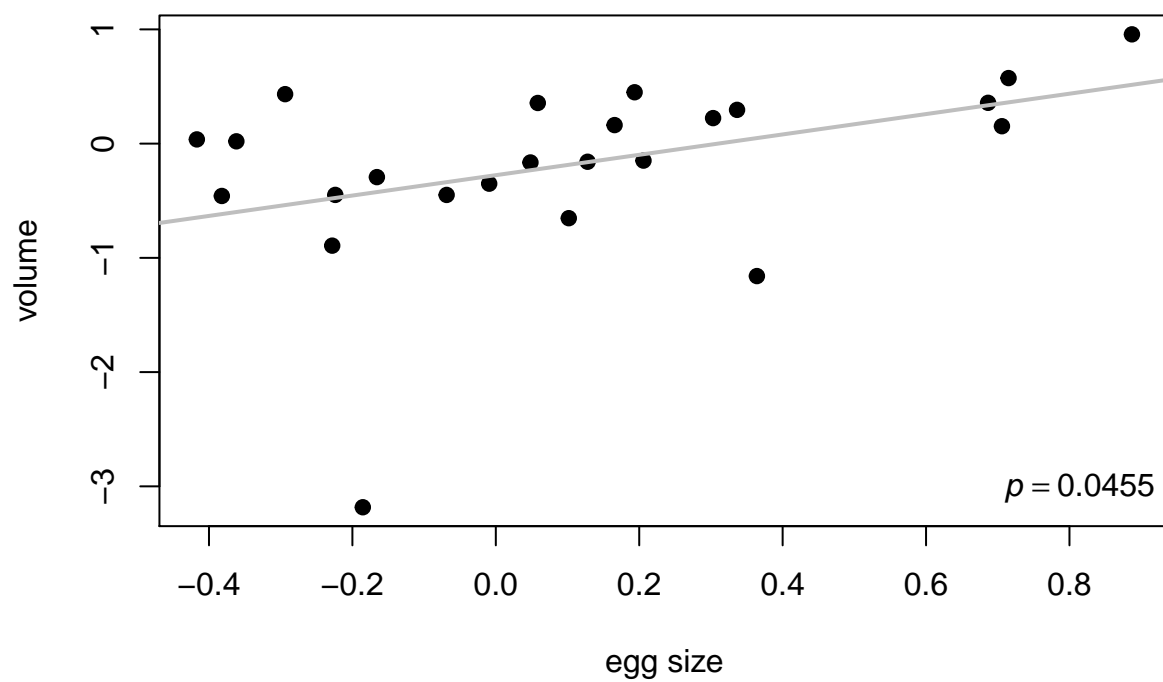
Body length male vs spermatocytesize

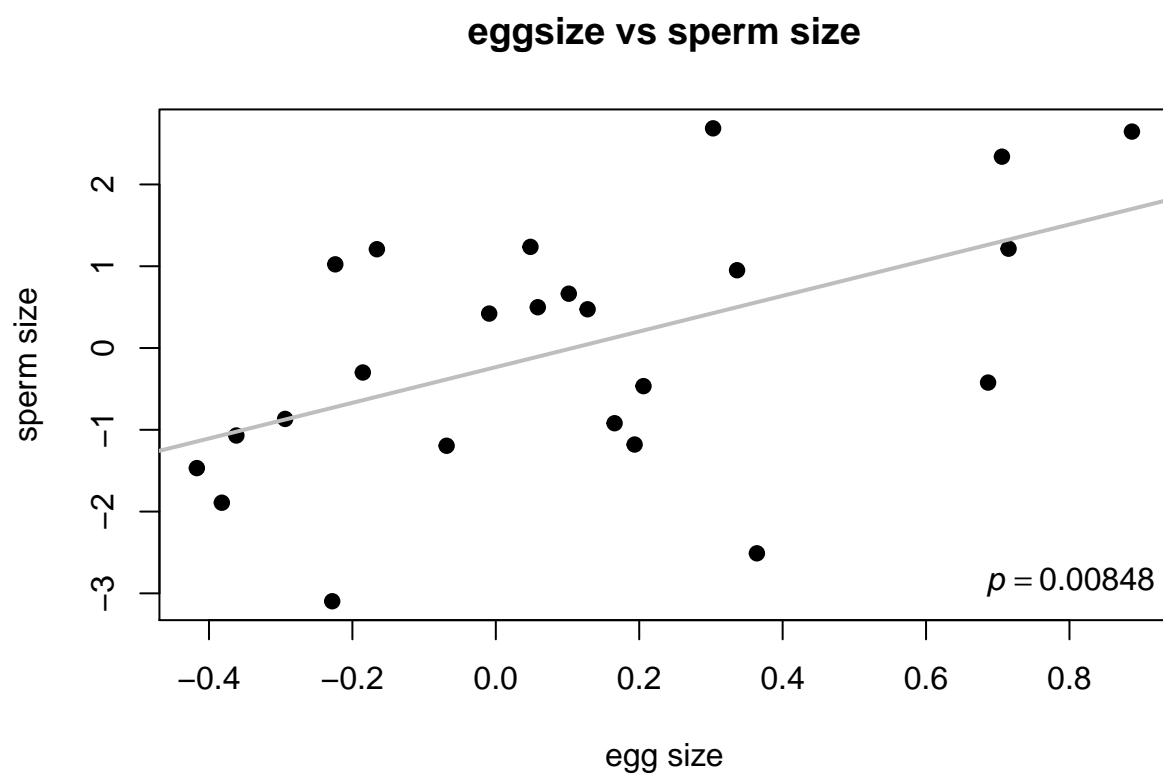


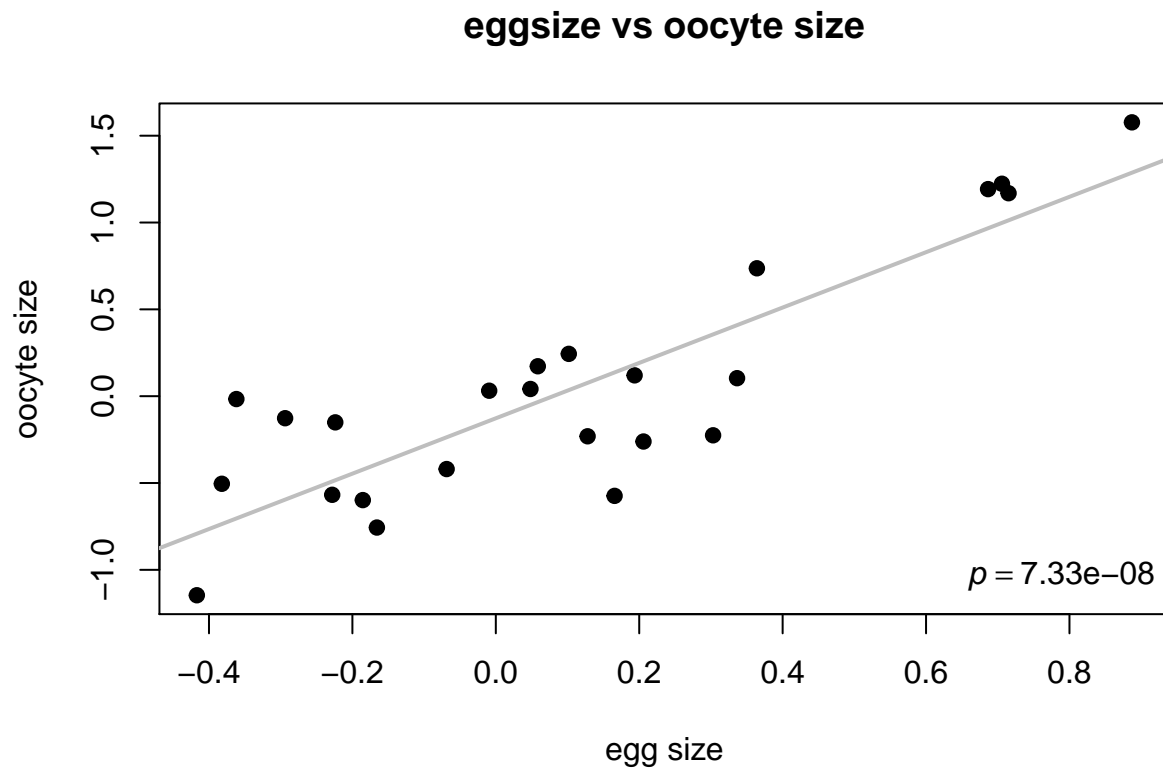
eggsize vs female length



eggsize vs female volume







PGLS - testing for correlation with evolutionary models

Output are summary tables from fits. The OU model won't fit the spermatocyte fits due to overparameterisation.

```
##
## Call:
## pgls(formula = meansperm ~ meanooarea, data = comp sperm, lambda = "ML")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.9643 -0.5988  0.4323  0.6861  1.8263
##
## Branch length transformations:
##
## kappa  [Fix]  : 1.000
## lambda [ML]  : 0.000
##   lower bound : 0.000, p = 1
##   upper bound : 1.000, p = 0.11287
##   95.0% CI   : (NA, NA)
## delta  [Fix]  : 1.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.09037    6.57159  -0.9268  0.3812
```

```

## meanooarea    1.46171    0.94975    1.5390    0.1624
##
## Residual standard error: 1.057 on 8 degrees of freedom
## Multiple R-squared: 0.2284, Adjusted R-squared: 0.132
## F-statistic: 2.369 on 1 and 8 DF, p-value: 0.1624

##
## Call:
## pgls(formula = meansperm ~ meanoo, data = compspem, lambda = "ML")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4586 -1.1960  0.6374  1.0989  1.4686
##
## Branch length transformations:
##
## kappa [Fix] : 1.000
## lambda [ ML] : 0.895
## lower bound : 0.000, p = 1
## upper bound : 1.000, p = 0.79696
## 95.0% CI    : (NA, NA)
## delta [Fix] : 1.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -11.31303    6.94067  -1.6300  0.14176
## meanoo       1.50685    0.65983   2.2837  0.05177 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.315 on 8 degrees of freedom
## Multiple R-squared: 0.3946, Adjusted R-squared: 0.319
## F-statistic: 5.215 on 1 and 8 DF, p-value: 0.05177

##
## Call:
## pgls(formula = meansperm ~ meanprimsp, data = compspem, lambda = "ML")
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.8244 -0.3185 -0.1789  0.1091  0.5614
##
## Branch length transformations:
##
## kappa [Fix] : 1.000
## lambda [ ML] : 0.000
## lower bound : 0.000, p = 1
## upper bound : 1.000, p = 0.0066141
## 95.0% CI    : (NA, 0.708)
## delta [Fix] : 1.000
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.63590    0.94589  -2.7867  0.0236793 *

```

```

## meanprimsp    1.21417    0.17131    7.0873 0.0001033 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.446 on 8 degrees of freedom
## Multiple R-squared:  0.8626,    Adjusted R-squared:  0.8454
## F-statistic: 50.23 on 1 and 8 DF,  p-value: 0.0001033

## Generalized least squares fit by REML
##   Model: meansperm ~ meanooarea
##   Data: fullpglsframe
##       AIC      BIC    logLik
##  54.95884 59.67106 -23.47942
##
## Correlation Structure: corMartins
## Formula: ~1
## Parameter estimate(s):
##   alpha
## 6.841132
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept) -3.044212  3.398811 -0.8956698  0.3793
## meanooarea   1.005070  0.498035  2.0180725  0.0549
##
## Correlation:
##      (Intr)
## meanooarea -0.999
##
## Standardized residuals:
##      Min      Q1      Med      Q3      Max
## -1.7775727 -0.6837636 -0.2886598  0.4237510  2.2862305
##
## Residual standard error: 0.6334179
## Degrees of freedom: 26 total; 24 residual

## Generalized least squares fit by REML
##   Model: meansperm ~ meanoo
##   Data: fullpglsframe
##       AIC      BIC    logLik
##  50.30285 55.01506 -21.15142
##
## Correlation Structure: corMartins
## Formula: ~1
## Parameter estimate(s):
##   alpha
## 5.469326
##
## Coefficients:
##              Value Std.Error   t-value p-value
## (Intercept) -7.068653  3.346328 -2.112361  0.0452
## meanoo       1.074887  0.329803  3.259179  0.0033
##
## Correlation:

```

```
##          (Intr)
## meanoo -0.999
##
## Standardized residuals:
##          Min          Q1          Med          Q3          Max
## -1.9084685 -0.7013875 -0.3474741  0.5038124  2.4034645
##
## Residual standard error: 0.5843052
## Degrees of freedom: 26 total; 24 residual
```

Some phylogenetic signal methods - lambda

```
## Loading required package: parallel
```

```
##      Length Class      Mode
## lik   1      bm      function
## bnd   2    data.frame list
## res 400    -none-    numeric
## opt   8    -none-    list
```

```
##      Length Class      Mode
## lik   1      bm      function
## bnd   2    data.frame list
## res 400    -none-    numeric
## opt   8    -none-    list
```

```
##      Length Class      Mode
## lik   1      bm      function
## bnd   2    data.frame list
## res 400    -none-    numeric
## opt   8    -none-    list
```

```
##      Length Class      Mode
## lik   1      bm      function
## bnd   2    data.frame list
## res 400    -none-    numeric
## opt   8    -none-    list
```

Blombergs K

```
##          K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.3738911          2.289904          3.964926          0.036
## PIC.variance.Z
## 1          -1.43097
```

```
##          K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.670685          0.4572932          1.216146          0.001
## PIC.variance.Z
## 1          -2.038024
```

```
## [1] "Dropping tips from the tree because they are not present in the data:"
## [1] "C. sp. 1"      "C. portoensis"  "C. virilis"
## [4] "C. sp. 8"      "C. angaria"     "C. castelli"
## [7] "C. sp. 2"      "C. kamaaina"    "C. imperialis"
## [10] "C. afra"       "C. nouraguensis" "C. doughertyi"
## [13] "C. wallacei"   "C. tropicalis"  "C. sp. 5"
## [16] "C. nigoni"
```

```
##          K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 1.218587      1.696502      2.587841      0.033
## PIC.variance.Z
## 1      -1.88925
```

```
##          K PIC.variance.obs PIC.variance.rnd.mean PIC.variance.P
## 1 0.5877787      0.2622437      0.6194661      0.002
## PIC.variance.Z
## 1      -1.744756
```

Morans I

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs Std.Obs Alter Pvalue
## 1 x -0.09396389 -1.460555 greater 0.944
##
## other elements: adj.method call
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs Std.Obs Alter Pvalue
## 1 x -0.01730539 0.6371791 greater 0.2
##
## other elements: adj.method call
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
```

```

##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs      Std.Obs  Alter Pvalue
## 1      x -0.1482137 -0.9646529 greater 0.84
##
## other elements: adj.method call

## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs      Std.Obs  Alter Pvalue
## 1      x -0.03082416 0.2752828 greater 0.326
##
## other elements: adj.method call

abouheifs cmean

## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs      Std.Obs  Alter Pvalue
## 1      x -0.09396389 -1.440322 greater 0.944
##
## other elements: adj.method call

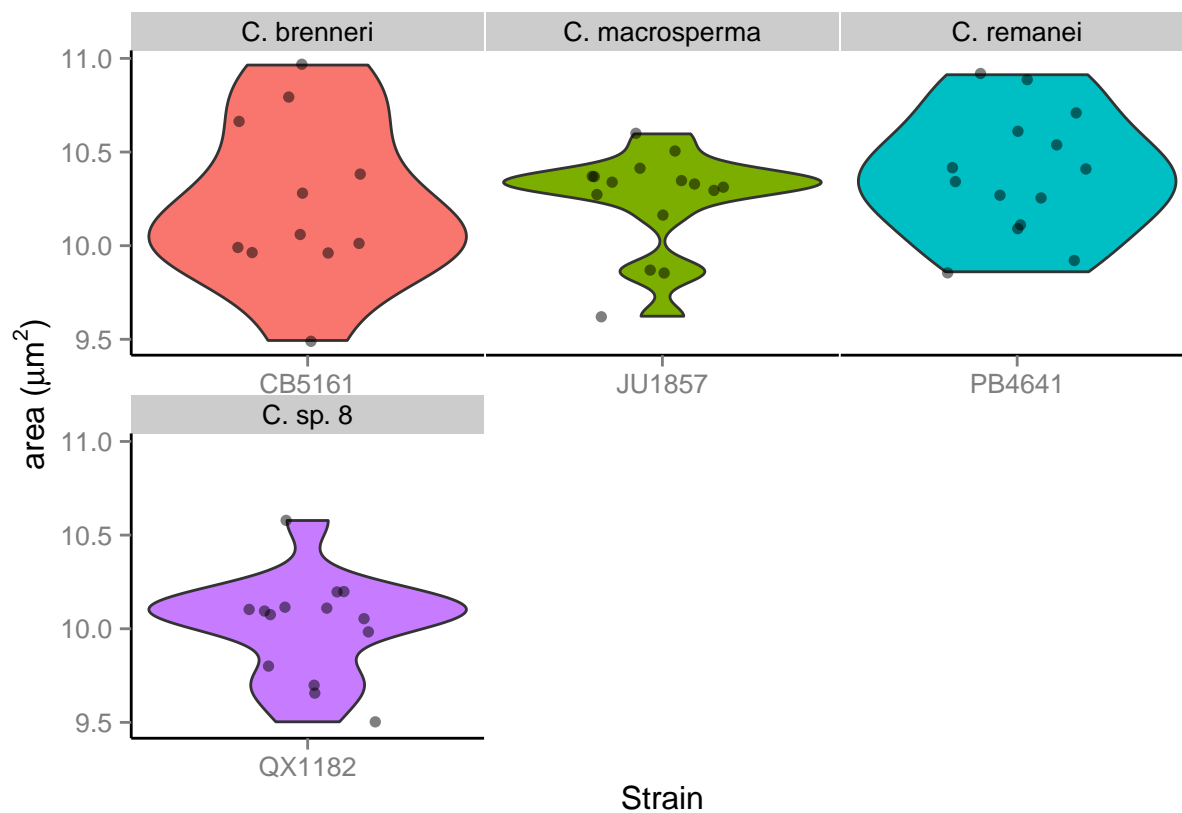
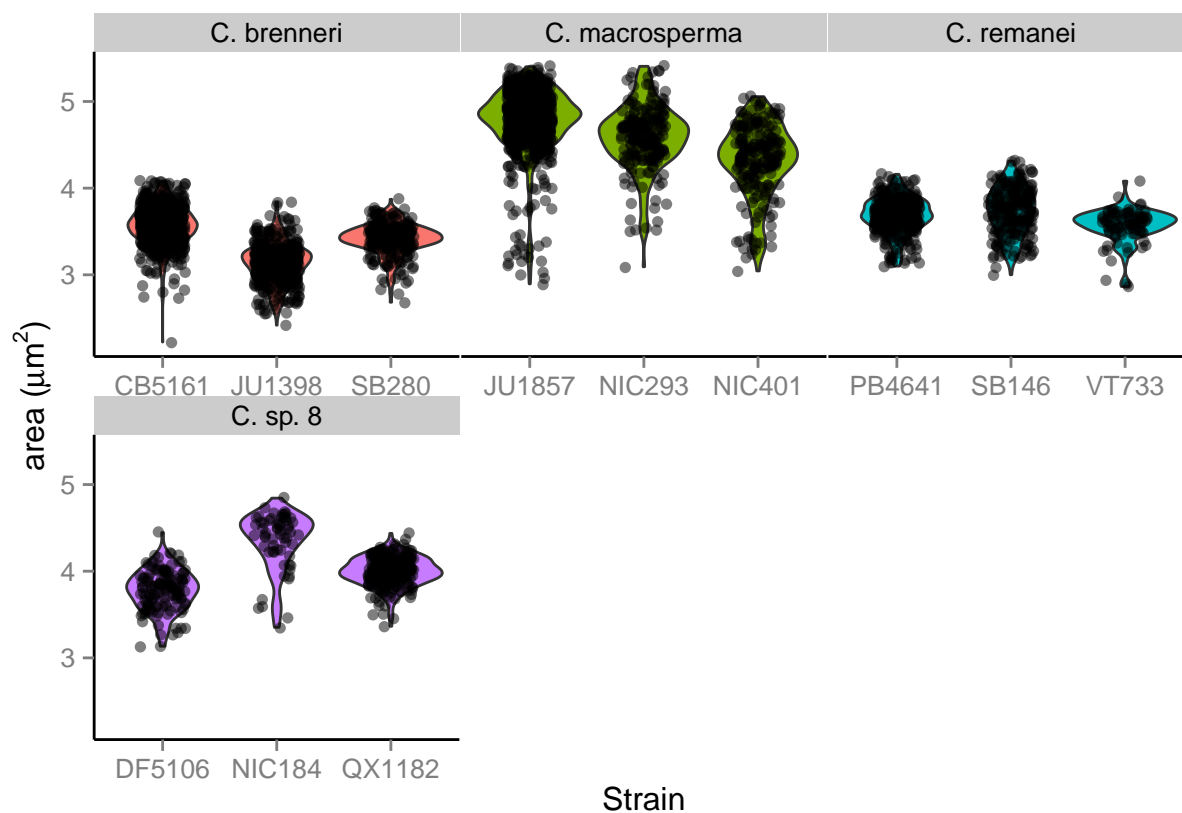
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs      Std.Obs  Alter Pvalue
## 1      x -0.01730539 0.6479682 greater 0.2
##
## other elements: adj.method call

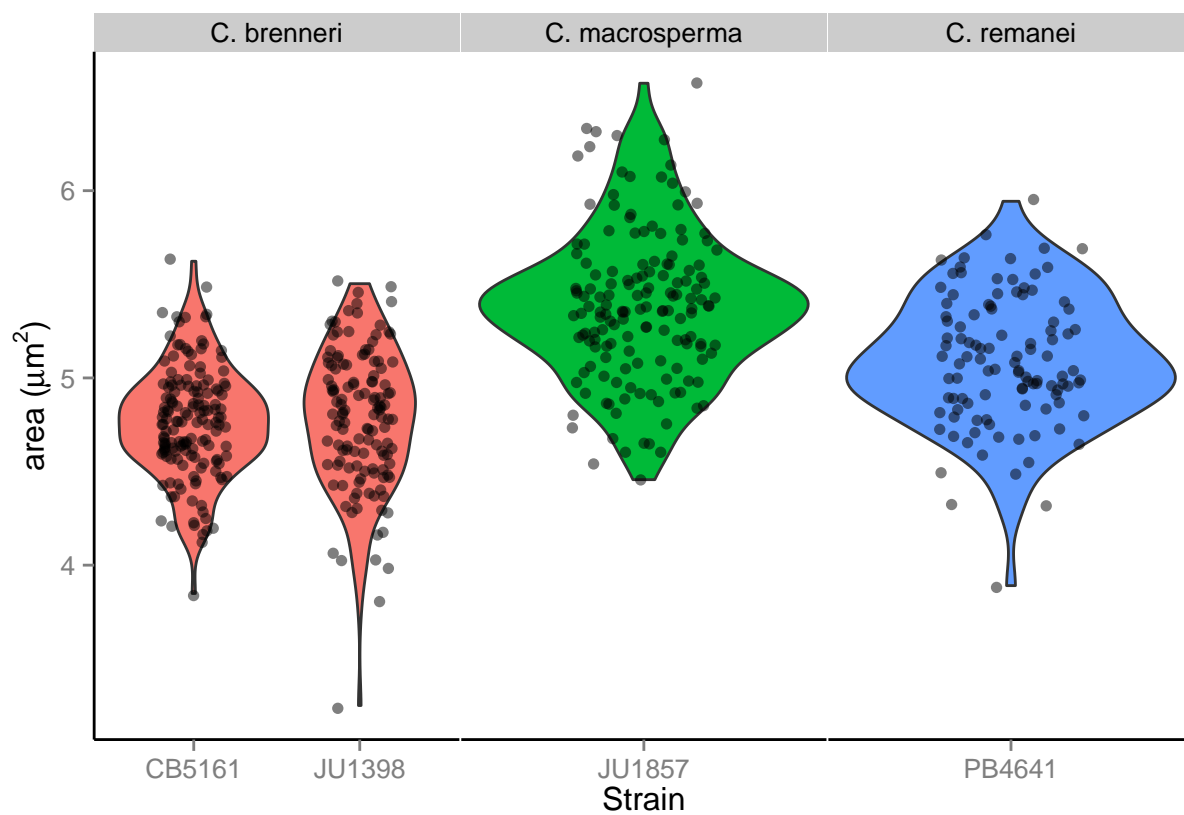
```

```
## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs      Std.Obs  Alter Pvalue
## 1      x -0.1482137 -0.9695812 greater 0.84
##
## other elements: adj.method call

## class: krandtest
## Monte-Carlo tests
## Call: as.krandtest(sim = matrix(res$result, ncol = nvar, byrow = TRUE),
##      obs = res$obs, alter = alter, names = test.names)
##
## Number of tests: 1
##
## Adjustment method for multiple comparisons: none
## Permutation number: 999
## Test      Obs      Std.Obs  Alter Pvalue
## 1      x -0.03082416 0.2915807 greater 0.324
##
## other elements: adj.method call
```

figure 3A with violin plots





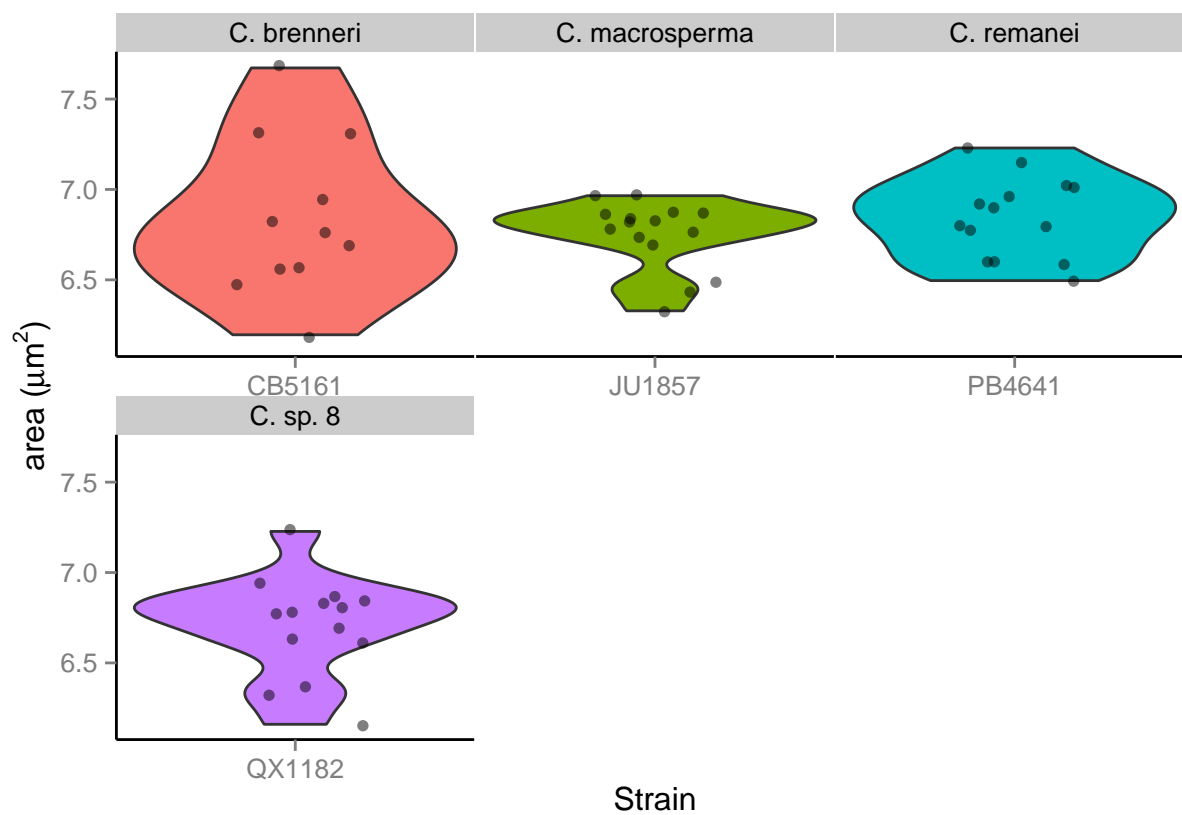
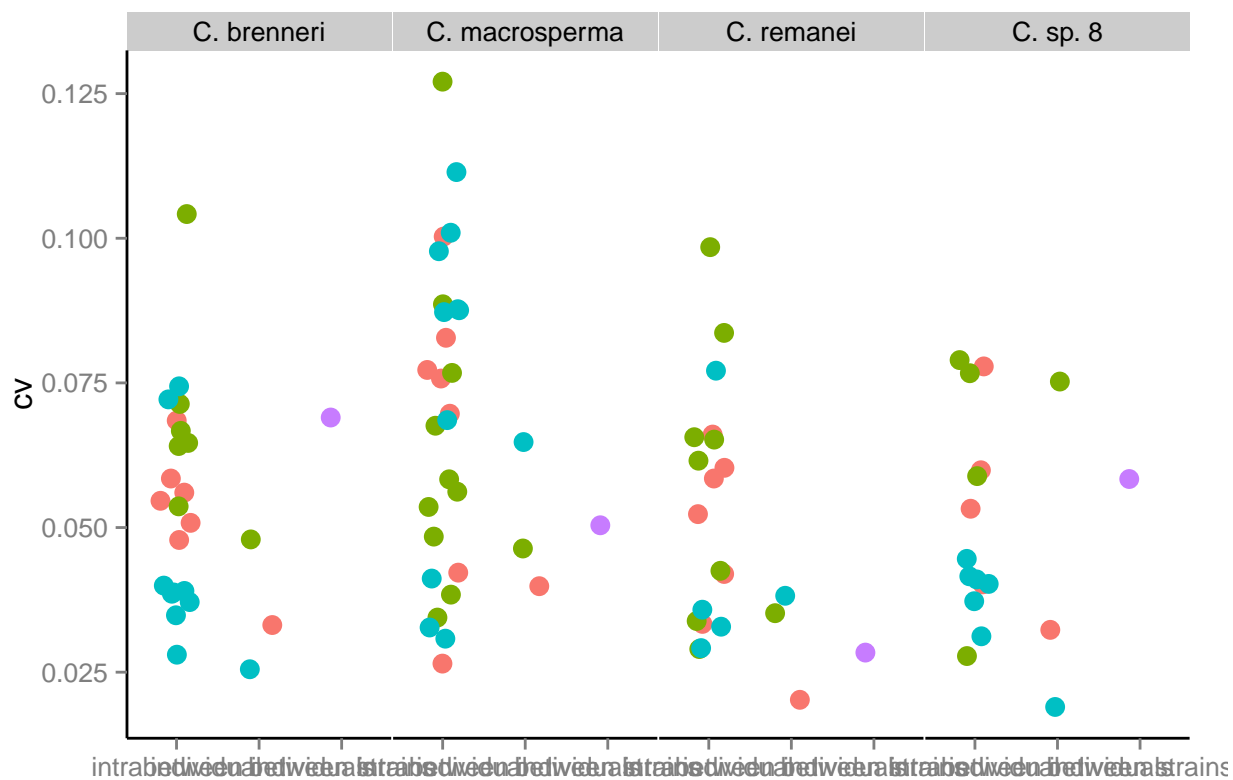
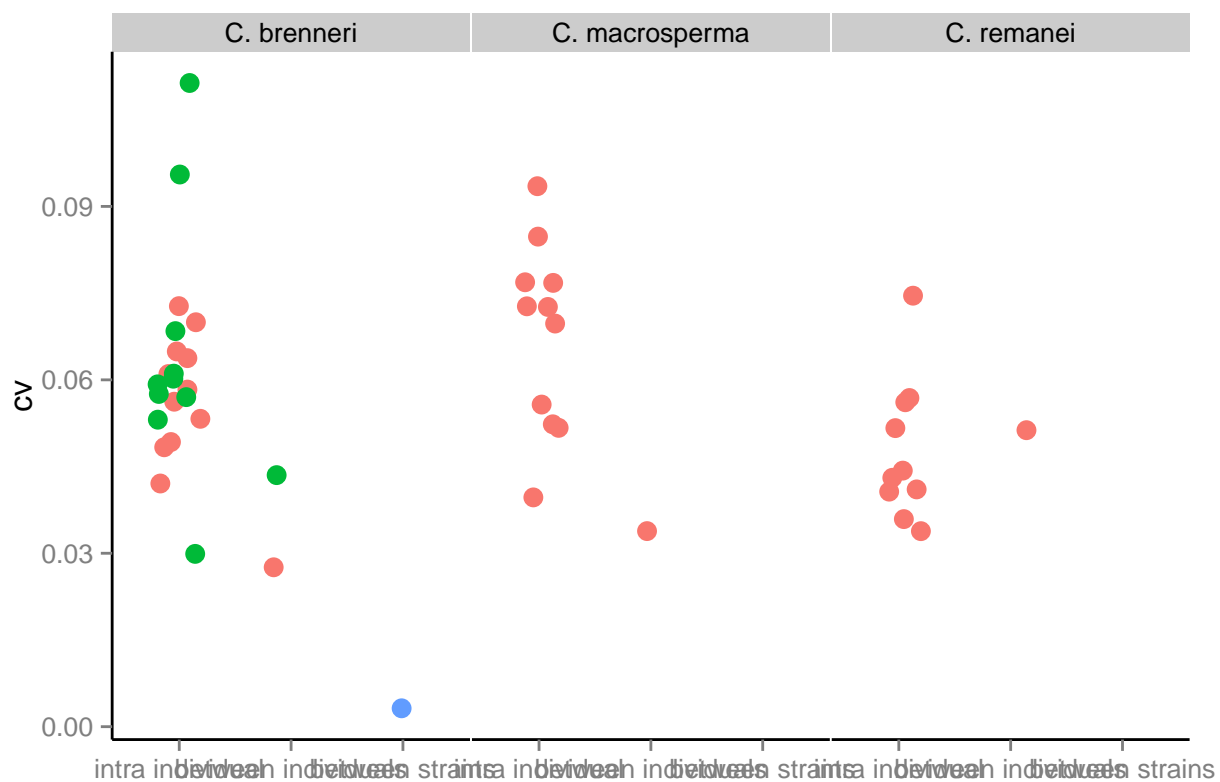
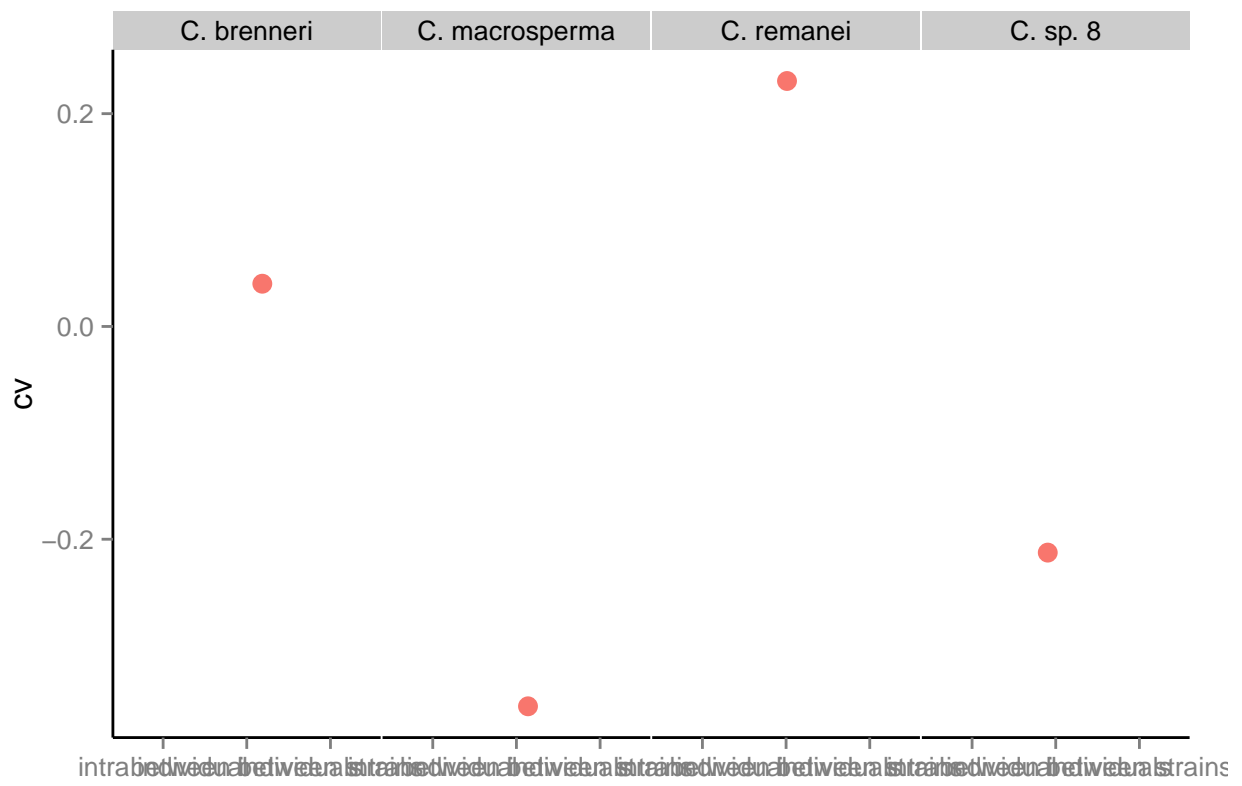


figure 3B





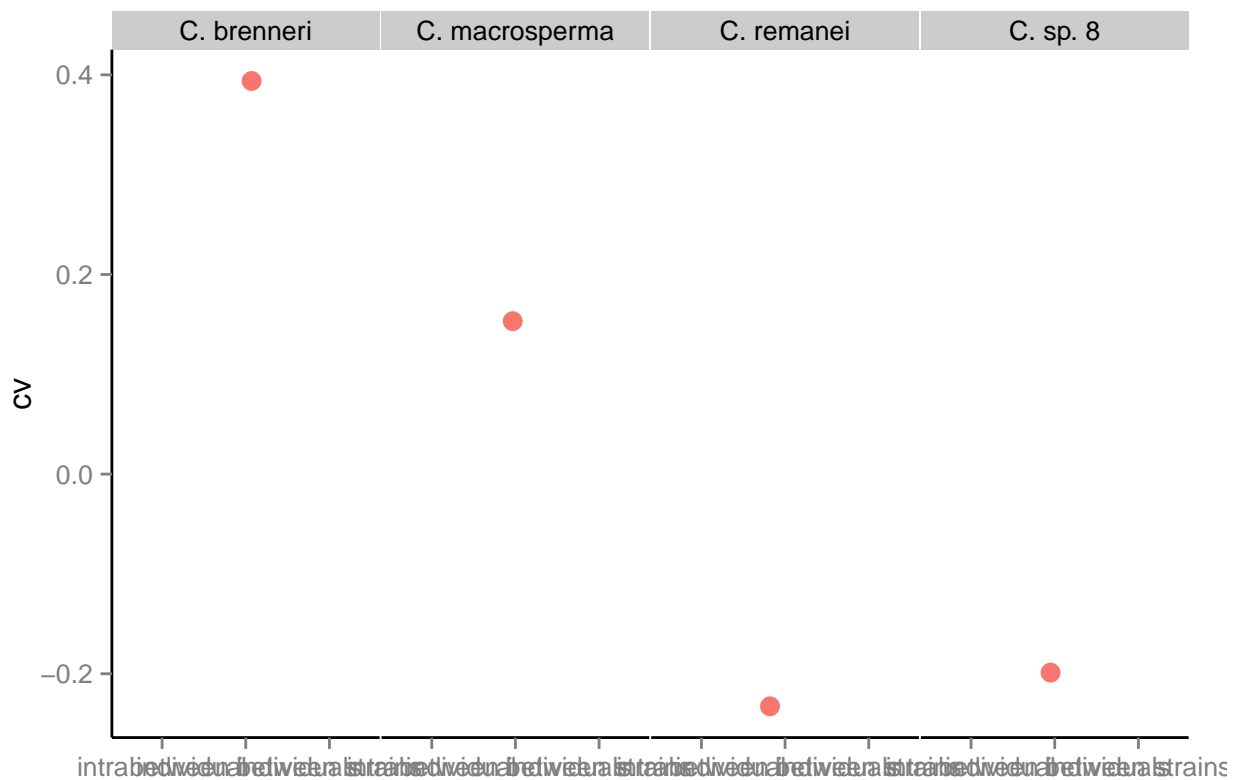


figure mockup a we are starting agin here, as I've modified some of the variables etc above Easier to start with a clean slate Clean and read in data:

mean and sd

```
groupedmeans=ff %>% group_by(Species,sex) %>% summarise(mean=mean(area,na.rm=T),n=n(),sd=sd(area,na.rm=T))
limits <- aes(ymax = mean + 1.96*sd/sqrt(n), ymin=mean - 1.96*sd/sqrt(n))
dodge <- position_dodge(width=0.9)
ggplot(data=groupedmeans, aes(x=Species, y=mean, fill=sex)) + geom_bar(stat="identity", position=position_dodge()) +
  scale_fill_manual(values=c("gray","white"))+ylab( expression(paste("area (", mu, m^{2},")")))+xlab(
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  panel.background = element_blank(),axis.line = element_line(colour = "black"),legend.position = "n
```

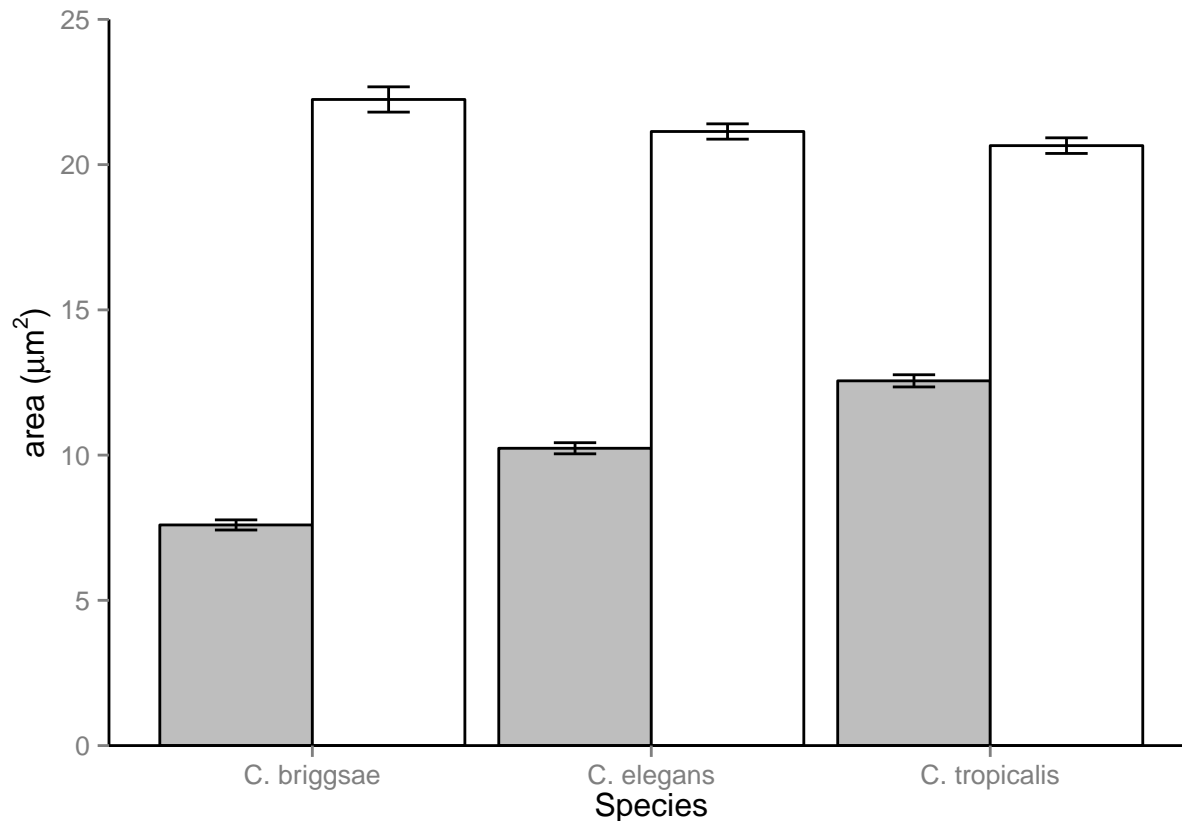


Figure mockup b group to get the data I want it's ugly, but it works

```
groupedstrains=ff %>% group_by(Species,Strain) %>%
  do(data.frame(meanM=mean(. [which(. $sex=="M"),]$area,na.rm=T),meanH=mean(. [which(. $sex=="H"),]$area,
    sdM=sd(. [which(. $sex=="M"),]$area,na.rm=T),sdH=sd(. [which(. $sex=="H"),]$area,na.rm=T))
groupedstrains$sem=1.96*groupedstrains$sdM/sqrt(groupedstrains$nM)
groupedstrains$seh=1.96*groupedstrains$sdH/sqrt(groupedstrains$nH)
p=ggplot(data=groupedstrains, aes(x=meanM, y=meanH, colour=Species, ymin = meanH - seh,ymax=meanH + seh))
p+geom_point(size=5)+coord_cartesian(ylim=c(0,15),xlim=c(0,33))+theme(panel.grid.major = element_blank(),
  panel.background = element_blank(),axis.line = element_line(colour = "black"),legend.position = "none")
p+ylab(expression(paste("Hermaphrodite Sperm area (", mu, m^{2},")")))+xlab(expression(paste("Male Sperm area (", mu, m^{2},")")))
p+geom_smooth(aes(group=Species), method="lm",size=1.5,colour="black")
```

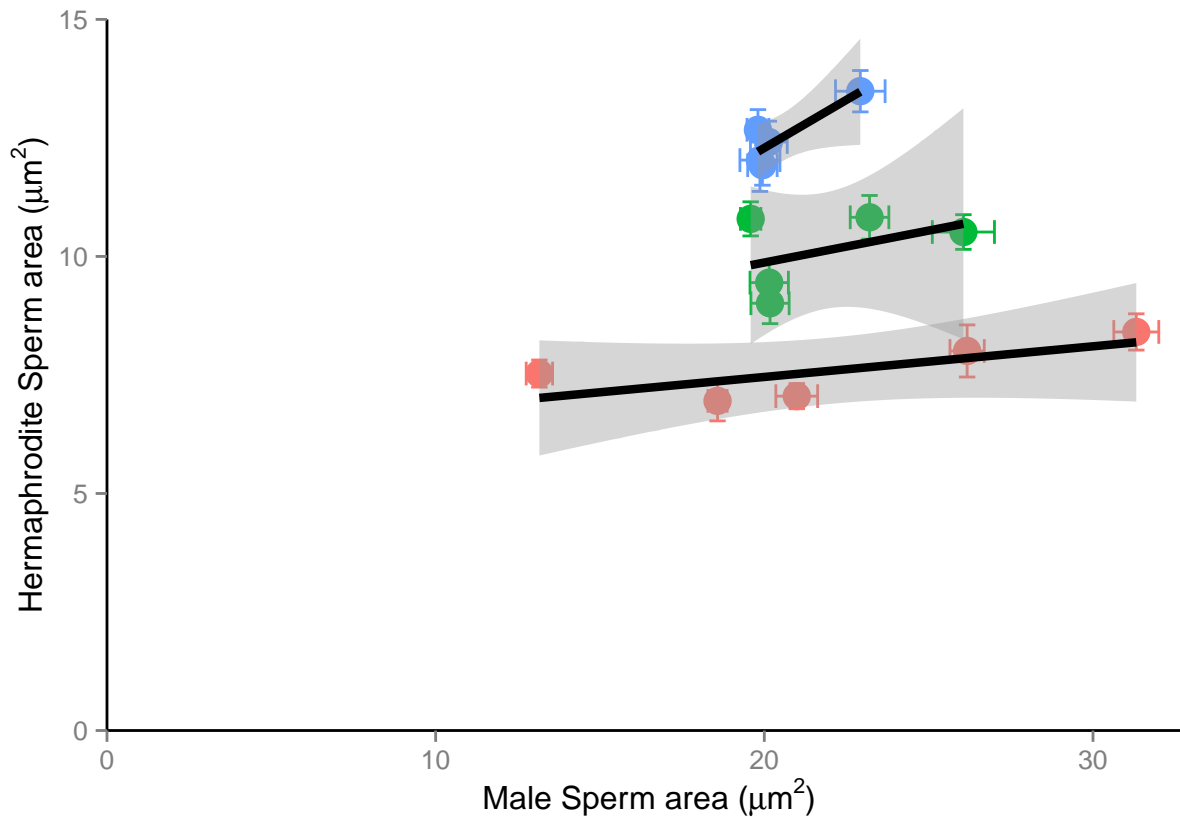


figure1a - bars need to rerun the above

```
ff<-spermsizes<-read.table("C:/Users/jeremy/Desktop/geiger/compileddata.csv",header=T,sep=",")
ff<-renameworms(ff)
spermsize2<-ff %>% group_by(Species,Strain) %>% summarise(means=mean(area,na.rm=T),cv=sd(area,na.rm=T)/
countspec=function(dataframe){
  holding=c()
  for(i in 1:nrow(dataframe)){
    holding=c(holding,sum(dataframe$Species==dataframe[i,]$Species))
  }
  return(holding)
}
spermsize2$Species<-ordered(spermsize2$Species,levels=c(tr$tip.label))
spermsize2<-spermsize2[order(spermsize2$Species),]
spermsize2$Strain<-factor(spermsize2$Strain, levels = spermsize2$Strain)
spermsize2$specnum=countspec(spermsize2)
spermsize2$cols=cut(spermsize2$means,3)
findchange<-function(df){
  holding=c()
  for(i in 1:length(df$Species)){
    if(i==1){
      holding=c(holding,0)
    } else if(df$Species[i]==df$Species[i-1]){
      holding=c(holding,0)
    } else {
      holding=c(holding,1)
    }
  }
}
```



```

    }
  }
  return(holding)
}

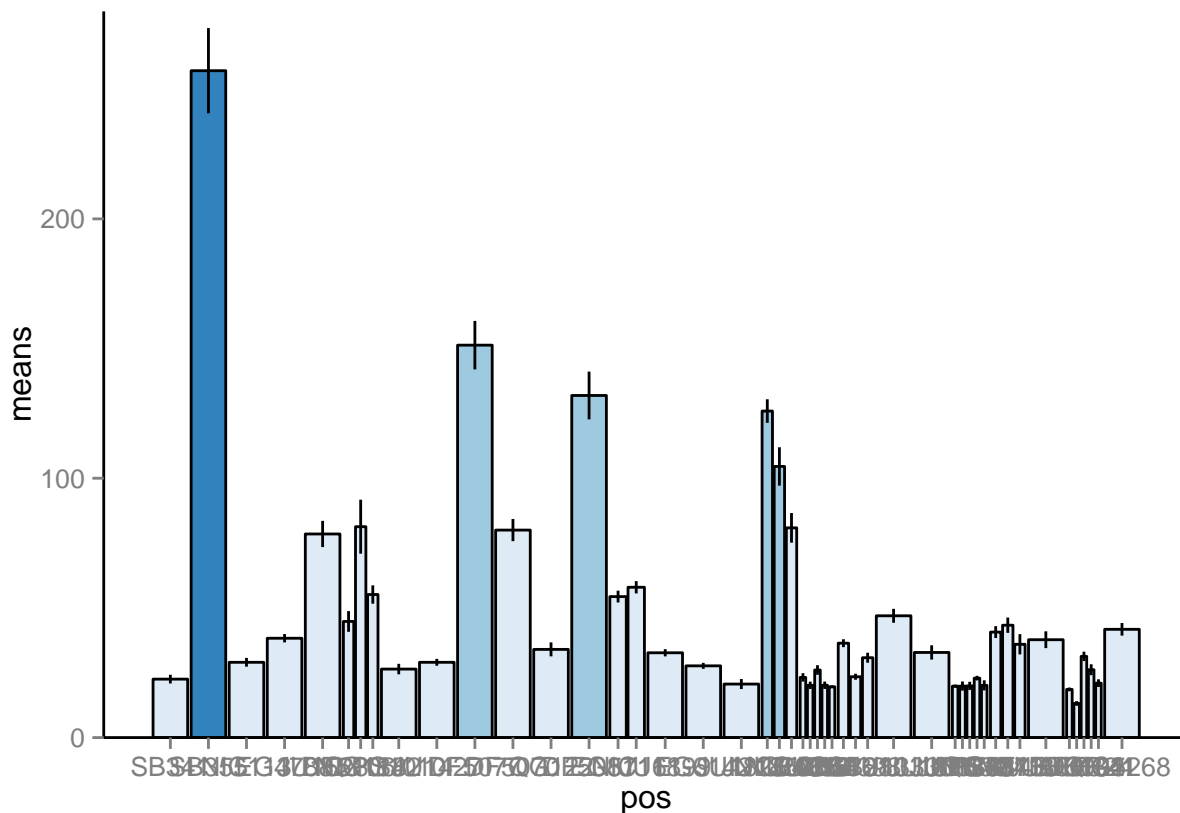
spermsize2$change<-findchange(spermsize2)
w<-(1-(spermsize2$specnum-1)*0.05)/spermsize2$specnum
pos <- 0.5 * (cumsum(w) + cumsum(c(0, w[-length(w)])))

gapsizes<-function(df,pos){
  pos1<-pos
  for(i in 1:length(df$specnum)){
    if(df$specnum[i]==1){
      pos1[i:length(pos1)]<-pos1[i:length(pos1)]+0.1
    } else if(df$change[i]==1){
      pos1[i:length(pos1)]<-pos1[i:length(pos1)]+0.1
    } else {
      pos1[i:length(pos1)]<-pos1[i:length(pos1)]+0.05
    }
  }
  return(pos1)
}

pos<-gapsizes(spermsize2,pos)
ggplot(data=spermsize2,aes(x = pos, width = w, y = means, fill=cols,ymin=means-sem,ymax=means+sem)) +
  geom_bar(stat = "identity",colour="black") + scale_x_continuous(labels = spermsize2$Strain, breaks =
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
    panel.background = element_blank(),axis.line = element_line(colour = "black"),legend.position = "n

```

Warning: position_stack requires constant width: output may be incorrect



large tree first group all the data

```
spermsize1<-ff %>% group_by(Species) %>% summarise(means=mean(area,na.rm=T))
spermsize2<-ff %>% group_by(Species,Strain) %>% summarise(means=mean(area,na.rm=T))
spermsize3<-ff %>% group_by(Species,Strain,individual) %>% summarise(means=mean(area,na.rm=T))
```

function to fix the format of spermsizes, and fit tree

```
fitace <- function(spermsizes,tree){
  holding<-spermsizes$means
  names(holding)<-spermsizes$Species
  return(fastAnc(tree,holding,CI=T))
}
```

function to choose a subset - works on any level, just chooses one from each species

```
subsetter<-function(spermsizes){
  x=lapply(split(spermsizes, spermsizes$Species),function(subdf) subdf[sample(1:nrow(subdf), 1),])
  do.call("rbind",x)
}
```

example usage fitace(subsetter(spermsizes2),tr) now need to loop and keep the values

```

looper<-function(spermsizes, reps, tree){
  outputdata=data.frame()
  outputCI=data.frame()
  for(i in 1:reps){
    outputdata=rbind(outputdata, fitace(subsetter(spermsizes), tree)$ace)
    outputCI=rbind(outputCI, fitace(subsetter(spermsizes), tree)$CI95)
  }
  names(outputdata)=seq(from=27, to=51)
  return(c(outputdata, outputCI))
}

```

ok we can call eg x=looper(spermsize3,10,tr) then mean and sd, etc. etc.

```

x<-fitace(spermsize1, tr)
AncSperm=x$ace
#or AncSperm=apply(x,2,mean) if it's bootstrapped
spermsize1$Species <- factor(spermsize1$Species, levels = c("C. sp. 1", "C. plicata", "C. guadeloupensis"))
spermsize=spermsize1[order(spermsize1$Species),]$means
names(spermsize)<-spermsize1[order(spermsize1$Species),]$Species
plotfancytree<-function(tree, nodesizes, terminalsizes){
  nodesizes<-nodesizes
  terminalsizes<-terminalsizes
  print(ggtree(tree)+geom_text(subset=.(isTip), aes(label=label), hjust=-0.19, fontface="italic")+
    geom_point(subset=.(!isTip), color="#1b9e77", size=sqrt(nodesizes/pi))+
    geom_point(subset=.(isTip), color="#7570b3", size=sqrt(terminalsizes/pi))+
    geom_text(subset=.(isTip), aes(label=round(terminalsizes,1)), hjust=1, vjust=-0.4, size=3)+
    geom_text(subset=.(!isTip), aes(label=round(nodesizes,1)), hjust=1.5, vjust=-0.4, size=3)+
    scale_x_continuous(expand = c(0.1, 0.1)))
  rm(list=c("terminalsizes", "nodesizes"), pos=".GlobalEnv")
}
plotfancytree(tr, AncSperm, spermsize)

```

