Final Draft

Kevin Montain

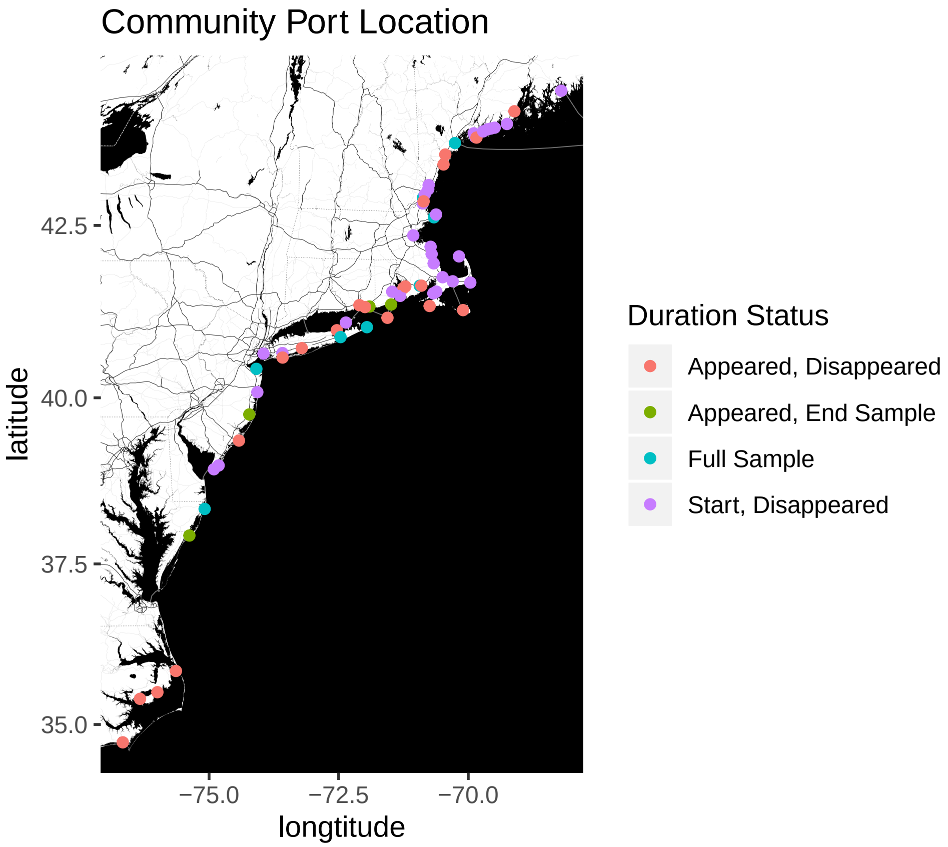
4/10/2020

Read the Data into R

```
setwd("/Users/kevin/Desktop/Aresty/")
dt <- read.csv("trips_data.csv")
dt <- setDT(dt)
```

Get the ports spatial positions

```
dt_ports <- dt %>% distinct(port_lat,port_lon,.keep_all = TRUE) %>% drop_na(.)
box <- make_bbox(dt_ports$port_lon,dt_ports$port_lat)
map <- get_stamenmap(box,maptype="toner-background",source='stamen',zoom=8,color='color',force=TRUE)
ggmap(map) + geom_point(aes(x = port_lon, y = port_lat, colour = duration.status), data = dt_ports) + la
```



Take each community over the years and get there ANN for first half of trips. Then average the ports that have survived the entire sample vs. those that haven't then look at those that survived longer than 15 years vs. those that didn't.

But first set up the cycle & the spatstat window

```r
### Create he Projection for analysis
lat = as.vector(c(dt$declat))
lon = as.vector(c(dt$declon))
xy = data.frame(lon, lat)
coordinates(xy) <- c("lon", "lat")
proj4string(xy) <- CRS("+proj=longlat +datum=NAD83")
NE <- spTransform(xy, CRS('+proj=laea +lat_0=45 +lon_0=-100 +x_0=0 +y_0=0 +a=6370997 +b=6370997 +units=
NE <- as.data.frame(NE)
dt <- cbind(dt,NE)
###Declare the window extent
W <- owin( c(min(dt$lon),max(dt$lon)), c(min(dt$lat), max(dt$lat)) )
comms <- as.vector(unique(dt$community))
## fish_ppp <- ppp(dt$declon,dt$declat,window = W)
```

Get the Survival Status per port

```r
comm_survstat <- dt %>% select(community, duration.status) %>% unique(.)
```

1) Cycle through the communities by community And by year +
2) Put the datatable into spatstat +
3) Calculate the Ann for first 100 neighbors. +
4) Put the neighbors into a precreated vector created with each iteration that is bound to a matrix with the previous neighbor count. +
5) Then calculate the average ANN per port and year.+
6) Put the result into a vector that will be added to the comm_survstat datatable.+

```r
### Initialize community level values
ann_result <- vector(length=length(comms))
comm <- vector(length = length(comms))
ind2 = 1

not_any_na <- function(x) all(!is.na(x)) ## Funct to clean NA values from future table
## Begin Community iterations
for (c in 1:length(comms)){

  ### Resetting values for each community **********
  ind1 = 1
  comm_years <- dt %>% select(community, year, lon, lat) %>% filter(community == comms[c])
  yrs <- length(unique(comm_years$year))
  splitPoint <- vector(length = yrs)
  ann_table <- as.data.table(matrix(seq(1:2500)))
  ### ************
  for (yr in 1:yrs){
    scratch <- comm_years %>% filter(year == sort(unique(comm_years$year))[yr])
    fish_ppp <- ppp(scratch$lon,scratch$lat,window = W,checkdup = FALSE)
    len = nrow(scratch)
    pt_dist <- floor(len/2)
    if (pt_dist < 2){next}
    else{
    ANN <- apply(nndist(fish_ppp, k=1:pt_dist),2,FUN=mean)
    ann_table <- cbind(ann_table,ANN)
    splitPoint[ind1] <- len
    ind1 = ind1 + 1} }

  ann_table <- ann_table[1:min(splitPoint)]
  colnames(ann_table) <- paste(1:ncol(ann_table))
  Filter(function(x)!all(is.na(x)), ann_table)
  ### iterate over the columns adding the vectors together into scr_col
  scr_col <- vector(length = nrow(ann_table))
  for (i in 2:ncol(ann_table)){
    scr_col <- scr_col + ann_table %>% pull(i)}
  ### Get the average across years and neighbors and provide to varaible
  value <- (scr_col[2]/ncol(ann_table-1))
  comm[[ind2]] <- comms[ind2]
  ann_result[[ind2]] <- value
  ind2 = ind2 + 1}

####
final_dt <- data.table(comm,ann_result)
colnames(final_dt) <- c("community","ANN")
comm_survstat = merge(x = comm_survstat, y = final_dt, by = "community", all.x = TRUE)
```

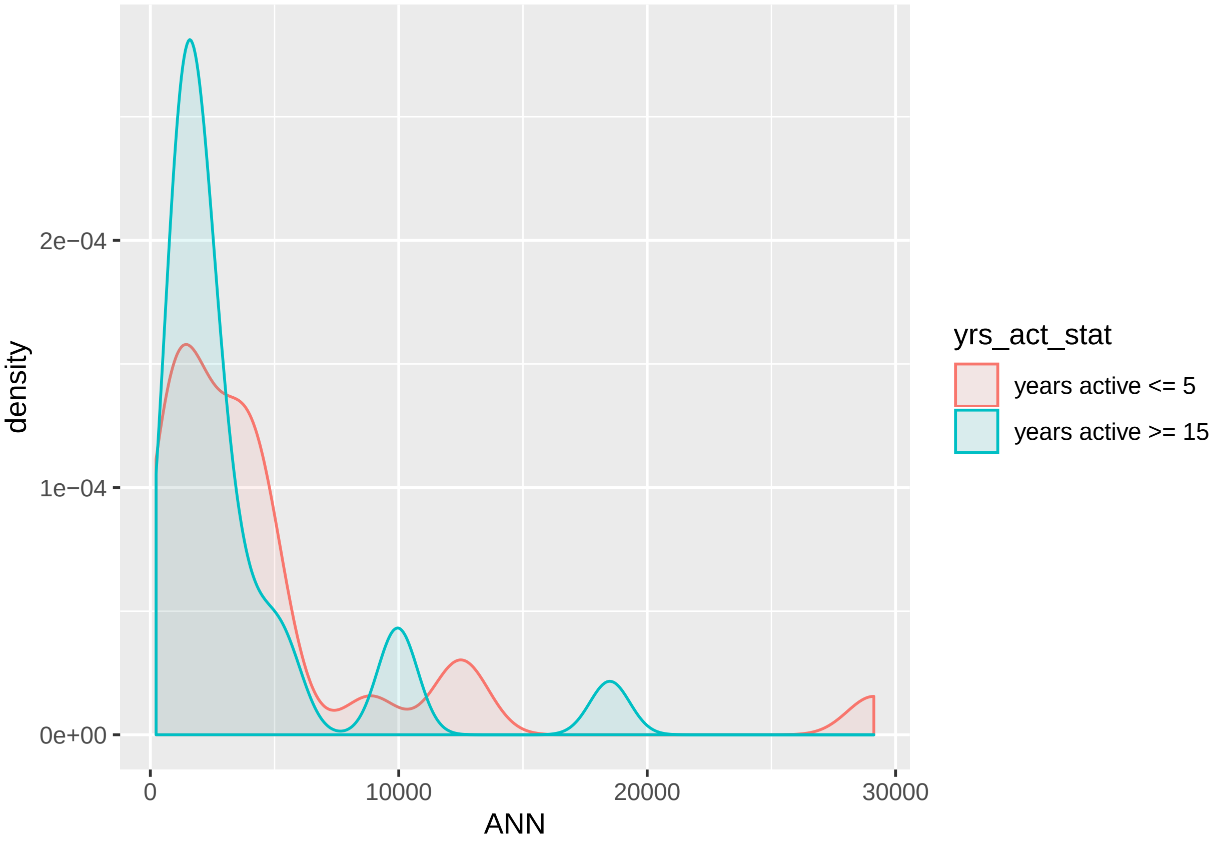
Take two groups, groups that lasted a long time >= 15 years and groups that lasted <= 5 years. Compare their means and variance. The following code will prepare the data and determine if it is normal or not. Normality will impact the variance calculation.

```
yrs_active <- vector(length = length(comms))
for (i in 1:length(comms)){
  scratch <- dt %>% filter(community == comms[i])
  yrs_active[i] <- length(unique(scratch$year))}
joinT <- as.data.table(cbind(comms, yrs_active))
colnames(joinT) <- c('community','yrs_active')
comm_years_stat = merge(x = comm_survstat, y = joinT, by = "community", all.x = TRUE)
t_table1 <- comm_years_stat[complete.cases(comm_years_stat),]
t_table1$yrs_active <- as.integer(t_table1$yrs_active)
full_sample <- t_table1 %>% filter(yrs_active >= 15)
no_full_sample <- t_table1 %>% filter(yrs_active <= 5)
d_set <- t_table1 %>% filter(yrs_active <= 5 | yrs_active>=15) %>% mutate(yrs_act_stat = ifelse(yrs_acti
## Is the data normal?
shapiro.test(d_set %>% filter(yrs_active <= 5) %>% pull(ANN))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  d_set %>% filter(yrs_active <= 5) %>% pull(ANN)
## W = 0.63516, p-value = 1.564e-06
```

The data is not normal. An F-test will not work. A T-test will work. A plot of the distribuition is:

```
ggplot(d_set,aes(ANN, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0.1)
```



Based on the Density Functions it may be best to use a Fligner-Killeen Test of Homogeneity of variances.

```
t.test(x = no_full_sample$ANN, y = full_sample$ANN)
```

```
##
##  Welch Two Sample t-test
##
## data:  no_full_sample$ANN and full_sample$ANN
## t = 0.72997, df = 40.23, p-value = 0.4696
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -1974.064  4206.914
## sample estimates:
## mean of x mean of y
##  4693.763  3577.339
```

```
fligner.test(ANN ~ yrs_act_stat,data=d_set)
```

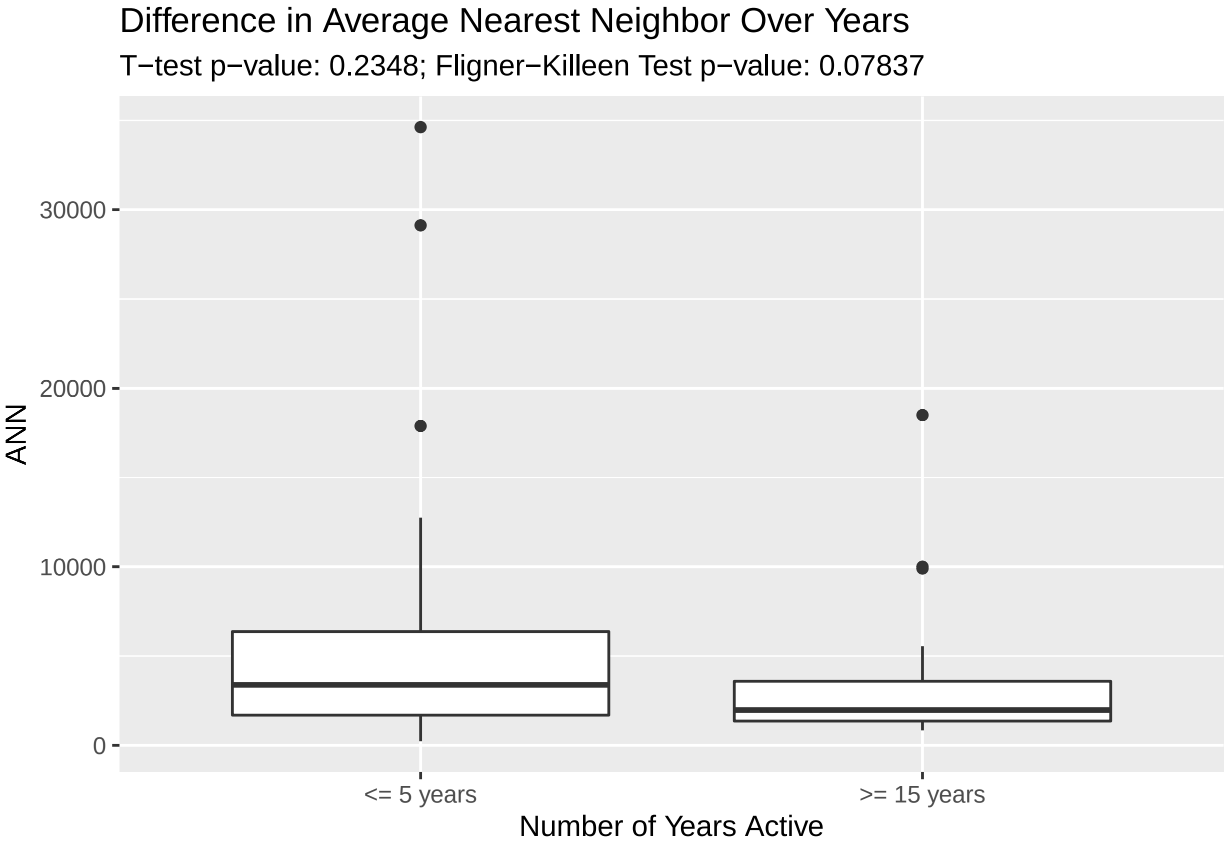
```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  ANN by yrs_act_stat
## Fligner-Killeen:med chi-squared = 3.0983, df = 1, p-value = 0.07837
```

```
leveneTest(ANN ~ yrs_act_stat,data=d_set,center=median,trim=.01)
```

```
## Levene's Test for Homogeneity of Variance (center = median: 0.01)
##       Df F value Pr(>F)
## group  1  0.6483 0.4249
##        45
```

I will create a box and whisker plot to show the composition of the data.

```
yrs_plot <- t_table1
yrs_plot <- yrs_plot %>% filter(yrs_active >= 15 || yrs_active <= 5)
yrs_title <- vector(length = nrow(yrs_plot))
for (i in 1:nrow(yrs_plot)){
  if (yrs_plot$yrs_active[i] >= 15){
    yrs_title[i] <- '>= 15 years'} else {yrs_title[i] <- '<= 5 years'}}
yrs_plot <- cbind(yrs_plot,yrs_title)
ggplot(data=yrs_plot, mapping=aes(x=yrs_title, y=ANN)) + geom_boxplot() + labs(x = "Number of Years Acti
```



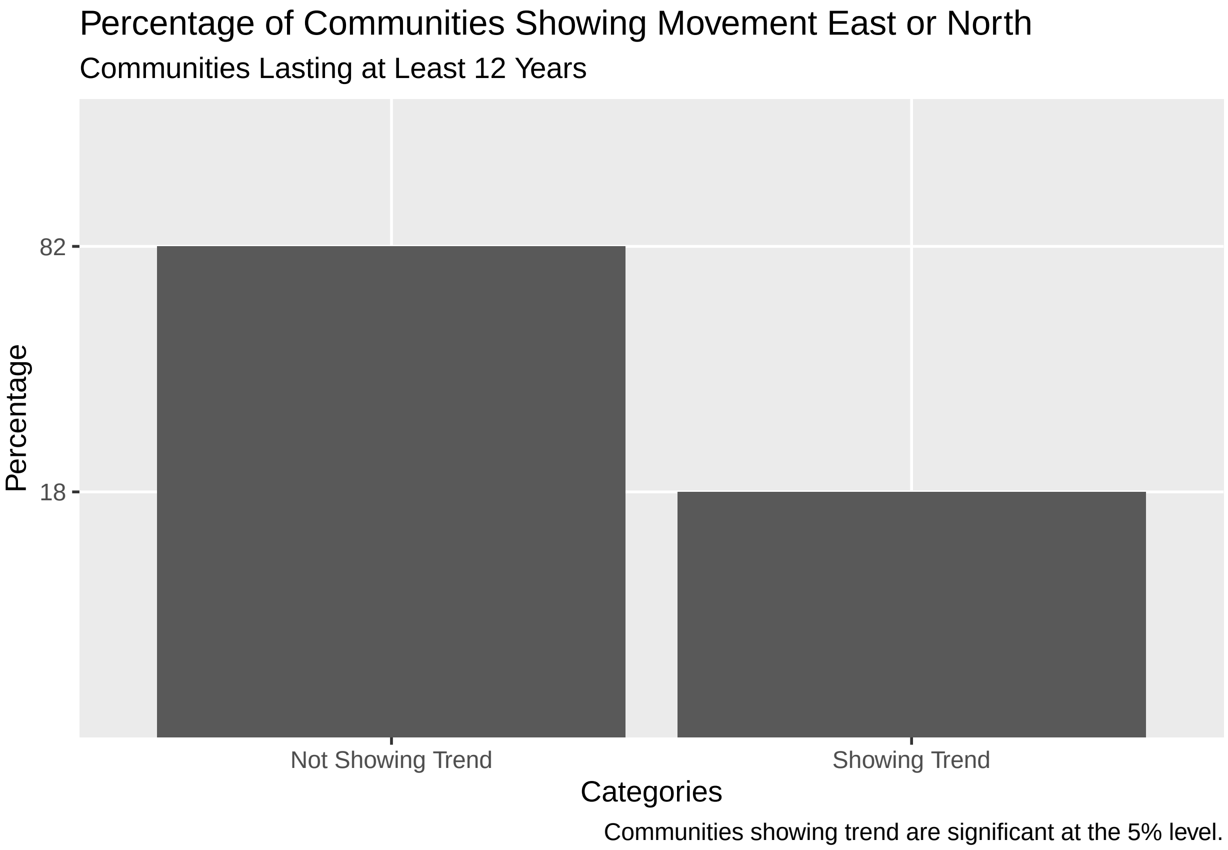
To get an analysis of community movement I will get the centroid of every year and the centroid of the first and second half of the sample. I will get the distance of the centroids. If the centroids are greater than 50 km and the differnces in lat/long means is greater than 1, meaning the second half is greater than the first sample, count the occurance. Look at the centroids and their distance apart

```
scratch1 <- left_join(x = dt, y = joinT, by = "community", all.x = TRUE)
scratch1$yrs_active <- as.double(scratch1$yrs_active)
scratch2 <- scratch1 %>% filter(yrs_active >= 12)
ct <- 0
for (co in 1:length(unique(scratch2$community))){
  scratch3 <- scratch2 %>% filter(community == unique(scratch2$community)[co])
  vals_above <- scratch3 %>% filter(year >= floor(mean(unique(year))) ) %>% distinct(year) %>% nrow()
  vals_below <- scratch3 %>% filter(year < floor(mean(unique(year))) ) %>% distinct(year) %>% nrow()
  below_thresh_x <- as.double(vector(length = vals_below))
  below_thresh_y <- as.double(vector(length = vals_below))
  above_thresh_x <- as.double(vector(length = vals_above))
```

```
 above_thresh_y <- as.double(vector(length = vals_above))
 ab <- 1
 be <- 1
 for (i in 1:length(unique(scratch3$year))){
  scratch4 <- scratch3 %>% filter(year == unique(scratch3$year)[i])
  W <- owin( c(min(scratch4$lon), max(scratch4$lon)), c(min(scratch4$lat), max(scratch4$lat) ))
  fish_ppp <- ppp(scratch4$lon,scratch4$lat, window = W, checkdup = FALSE)
  if (scratch4$year[1] < floor(mean(unique(scratch3$year)))) {
   below_thresh_x[[be]] <- as.double(centroid.owin(W, as.ppp = FALSE)[1])
   below_thresh_y[[be]] <- as.double(centroid.owin(W, as.ppp = FALSE)[2])
   be = be + 1}
  else{
   above_thresh_x[[ab]] <- as.double(centroid.owin(W, as.ppp = FALSE)[1])
   above_thresh_y[[ab]] <- as.double(centroid.owin(W, as.ppp = FALSE)[2])
   ab = ab + 1} }
  above_W <- owin( c(min(above_thresh_x), max(above_thresh_x)), c(min(above_thresh_y), max(above_thres
  above_centroid_ppp <- ppp(above_thresh_x,above_thresh_y, window = above_W, checkdup = FALSE)
  above_centroid <- centroid.owin(above_W, as.ppp = TRUE)
  cord_above <- as.double(centroid.owin(above_W, as.ppp = FALSE))
  below_W <- owin( c(min(below_thresh_x), max(below_thresh_x)), c(min(below_thresh_y), max(below_thres
  below_centroid_ppp <- ppp(below_thresh_x,below_thresh_y, window = below_W, checkdup = FALSE)
  below_centroid <- centroid.owin(below_W, as.ppp = TRUE)
  cord_below <- as.double(centroid.owin(below_W, as.ppp = FALSE))
  aboveBelow_dist <- crossdist(below_centroid, above_centroid)[1]/1000
  if ( ((cord_below[1] < cord_above[1])|(cord_below[2] < cord_above[2])) & (cord_above[2] > cord_belo
   ### x's are bigger
   if ((cord_below[1] < cord_above[1])){
    p <- as.double(t.test(above_thresh_x, below_thresh_x, alternative = 'g')[3])
    if (p < .05){ct <- ct + 1} }


   ### y's are bigger
   if ((cord_below[2] < cord_above[2])){
    p <- as.double(t.test(above_thresh_y, below_thresh_y, alternative = 'g')[3])
    if (p < .05){ct <- ct + 1} }
   ### Both are bigger
   if ((cord_below[1] < cord_above[1])){
    p1 <- as.double(t.test(above_thresh_x, below_thresh_x, alternative = 'g')[3])
    p2 <- as.double(t.test(above_thresh_x, below_thresh_x, alternative = 'g')[3])
    if (p1 < .05 | p2 < .05){ct <- ct + 1} } } }
```

I will produce a plot showing the count of significant movement

```
a <- round(ct/length(unique(scratch2$community))*100,0)
b <- round(100 - a,0)
vals <- c(a,b)
cats <- c('Showing Trend','Not Showing Trend')
ma <- as.data.table(cbind(cats,vals))
ggplot(ma, aes(cats,vals)) + geom_col() + labs(title="Percentage of Communities Showing Movement East o
```



Now I am going to go back to the main dt brought into this r session. 1) Get the community boat count by year 2) Create a new dt that aggregates data by community and year 3) merge the two data tables 4) Then create a logit model.

```
### 1)
yr <- vector()
cm <- vector()
bts <- vector()
counter <- 1
for (i in 1:length(unique(dt$community))){
  tb <- dt %>% filter(community == unique(dt$community)[i])
  for (j in 1:length(unique(tb$year))){
    scratch <- tb %>% filter(year == unique(tb$year)[j])
    bts[[counter]] <- scratch %>% nrow()
    yr[[counter]] <- unique(tb$year)[j]
    cm[[counter]] <- as.character(unique(dt$community)[i])
    counter = counter + 1} }
tbs <- data.table(cbind(cm,as.character(yr),as.character(bts)))
colnames(tbs) <- c('community','year','ct')
tbs$year <- as.integer(tbs$year)
tbs$count <- as.integer(tbs$count)
### 2)
n <- dt
n$community <- as.character(n$community)
joinT$yrs_active <- as.integer(joinT$yrs_active)
new_dt <- left_join(x = n, y = joinT, by = "community", all.x = TRUE)
new_dt <- new_dt %>% select(community,qtykept,adj_fisher_days,dist_from_port,richness,yrs_active,year,
### 3)
table <- left_join(x = as.data.table(new_dt), y = tbs, by = c("community",'year'), all.x = TRUE)
```

Now create the equation using variables, step 4

```
table$ct <- as.integer(table$ct)
table$yearfol_survival <- as.integer(table$yearfol_survival)
### Start with a Binomial
mdl <- glm(yearfol_survival ~ richness + log(ct, base = exp(2)) + qtykept + adj_fisher_days + dist_from_
print(pR2(mdl))
```

```
## fitting null model for pseudo-r2
##          llh      llhNull           G2     McFadden         r2ML         r2CU
## -203.9286656 -258.3137647  108.7701983    0.2105389    0.1446795    0.2761201
```

```
summary(mdl)
```

```
##
## Call:
## glm(formula = yearfol_survival ~ richness + log(ct, base = exp(2)) +
##     qtykept + adj_fisher_days + dist_from_port, family = binomial("logit"),
##     data = table)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4645   0.1585   0.3123   0.4987   1.8643
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.784e+00  6.246e-01  -4.458 8.29e-06 ***
## richness                3.575e-02  1.838e-02   1.945   0.0518 .
## log(ct, base = exp(2))  1.573e+00  3.230e-01   4.871 1.11e-06 ***
## qtykept                 1.732e-06  5.319e-06   0.326   0.7446
## adj_fisher_days        -2.891e-02  2.624e-02  -1.102   0.2705
## dist_from_port          8.026e-03  4.239e-03   1.894   0.0583 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 516.63  on 695  degrees of freedom
## Residual deviance: 407.86  on 690  degrees of freedom
##   (15 observations deleted due to missingness)
## AIC: 419.86
##
## Number of Fisher Scoring iterations: 6
```

Based on McFadden this is an excellent model of survivability. Good models fit in the range of .2-.4

Now take the data and aggregate the data to the community level.

```
n <- dt
n$community <- as.character(n$community)
joinT$yrs_active <- as.integer(joinT$yrs_active)
new_dt <- left_join(x = n, y = joinT, by = "community", all.x = TRUE)
table2 <- left_join(x = as.data.table(new_dt), y = tbs, by = c("community",'year'), all.x = TRUE)
table2$ct <- as.integer(table2$ct)
```

```
ttTable <- table2 %>% select(community,qtykept,ct,adj_fisher_days,dist_from_port,richness,yrs_active,yea
mergeTable <- dt %>% select(duration.status,community) %>% unique(.)
ttTable <- left_join(x = ttTable, y = mergeTable, by = "community", all.x = TRUE)
```

Now take the ttest of each column

```
sig_t <- vector()
ct_t <- 1
for (i in 2:ncol(ttTable)){
  if (i>=7){next}
  else{
    groups <- ttTable %>% filter(yrs_active <= 5 | yrs_active >= 15) %>% mutate(yrs_act_stat = ifelse(y
    top_group <- groups %>% filter(yrs_active >= 15) %>% pull(i)
    bottom_group <- groups %>% filter(yrs_active <= 5) %>% pull(i)
    t_p <- as.double(t.test(top_group, bottom_group, alternative='g')[3])
    if (t_p < .05){sig_t[[ct_t]] <- colnames(groups)[i]
    ct_t = ct_t + 1} } }
print(paste("Characterisitcs where the older ports have larger means",sig_t))
```

```
## [1] "Characterisitcs where the older ports have larger means ct"
## [2] "Characterisitcs where the older ports have larger means richness"
```

Now Following from above observe the normality of the data. Then plot the data for each column. If the data is very skewed use Fligner-Killeen if not use the levine test. But first get the column names.

```
groups <- ttTable %>% filter(yrs_active <= 5 | yrs_active >= 15) %>% mutate(yrs_act_stat = ifelse(yrs_a
colnames(groups)
```

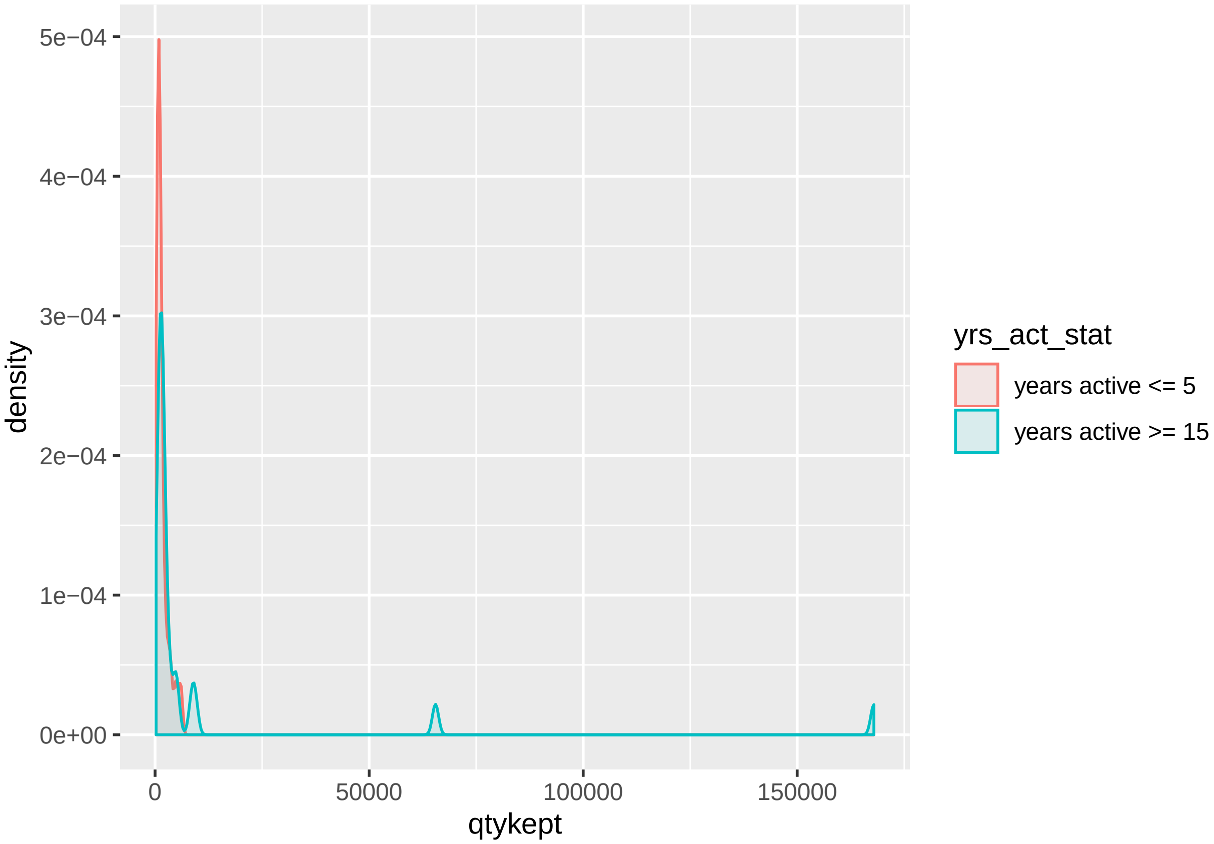
```
##  [1] "community"       "qtykept"         "ct"              "adj_fisher_days"
##  [5] "dist_from_port"  "richness"        "yrs_active"      "year"
##  [9] "duration.status" "yrs_act_stat"
```

Get the shapiro test of quantity kept and create the density plot.

```
shapiro.test(groups %>% filter(yrs_active >= 15) %>% pull(qtykept))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  groups %>% filter(yrs_active >= 15) %>% pull(qtykept)
## W = 0.3395, p-value = 1.354e-09
```

```
ggplot(groups,aes(qtykept, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0.1)
```



Hard to determine if the Levene of flinger test should be used so I will use both.

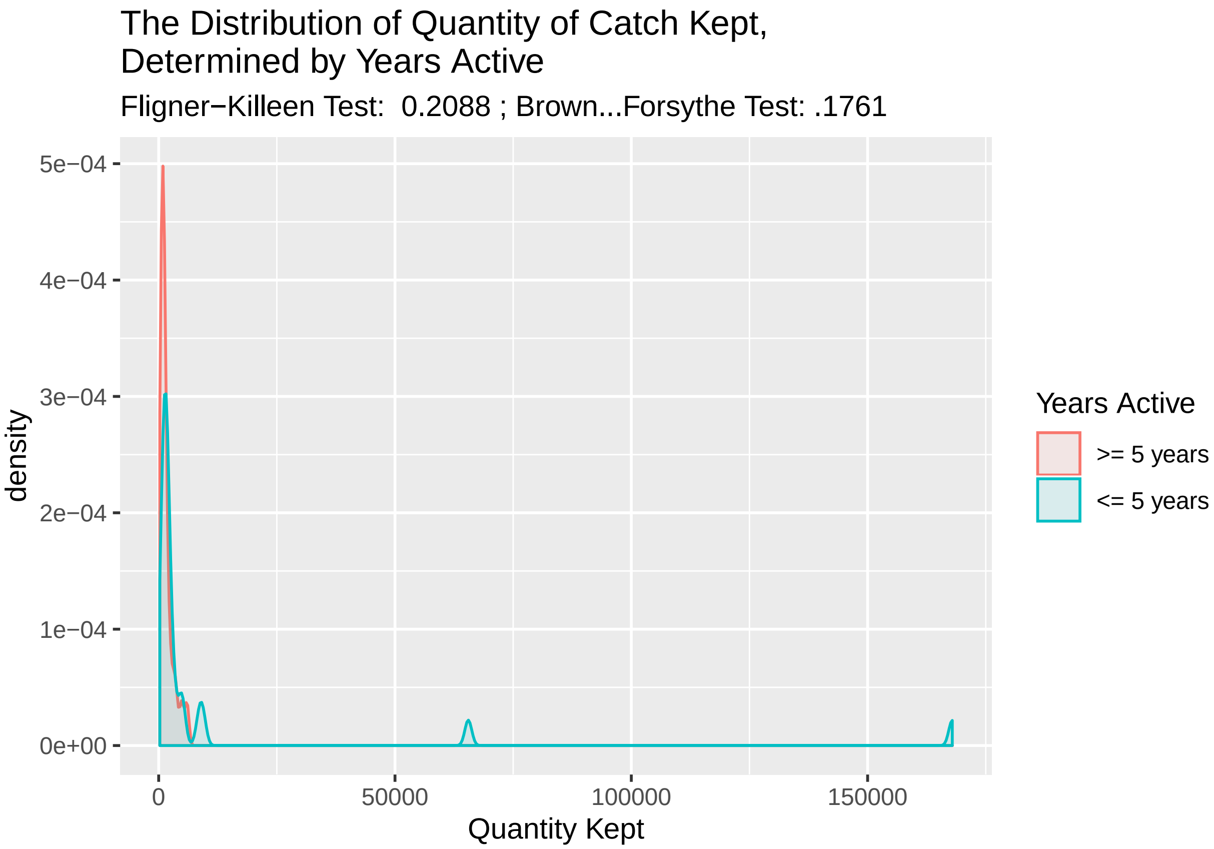
```
fligner.test(qtykept ~ yrs_act_stat, groups)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  qtykept by yrs_act_stat
## Fligner-Killeen:med chi-squared = 1.5796, df = 1, p-value = 0.2088
```

```
leveneTest(qtykept ~ yrs_act_stat, groups, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1   1.885 0.1761
##       48
```

```
ggplot(groups,aes(qtykept, fill = yrs_act_stat, colour = yrs_act_stat)) + geom_density(alpha = 0.1) + s
```



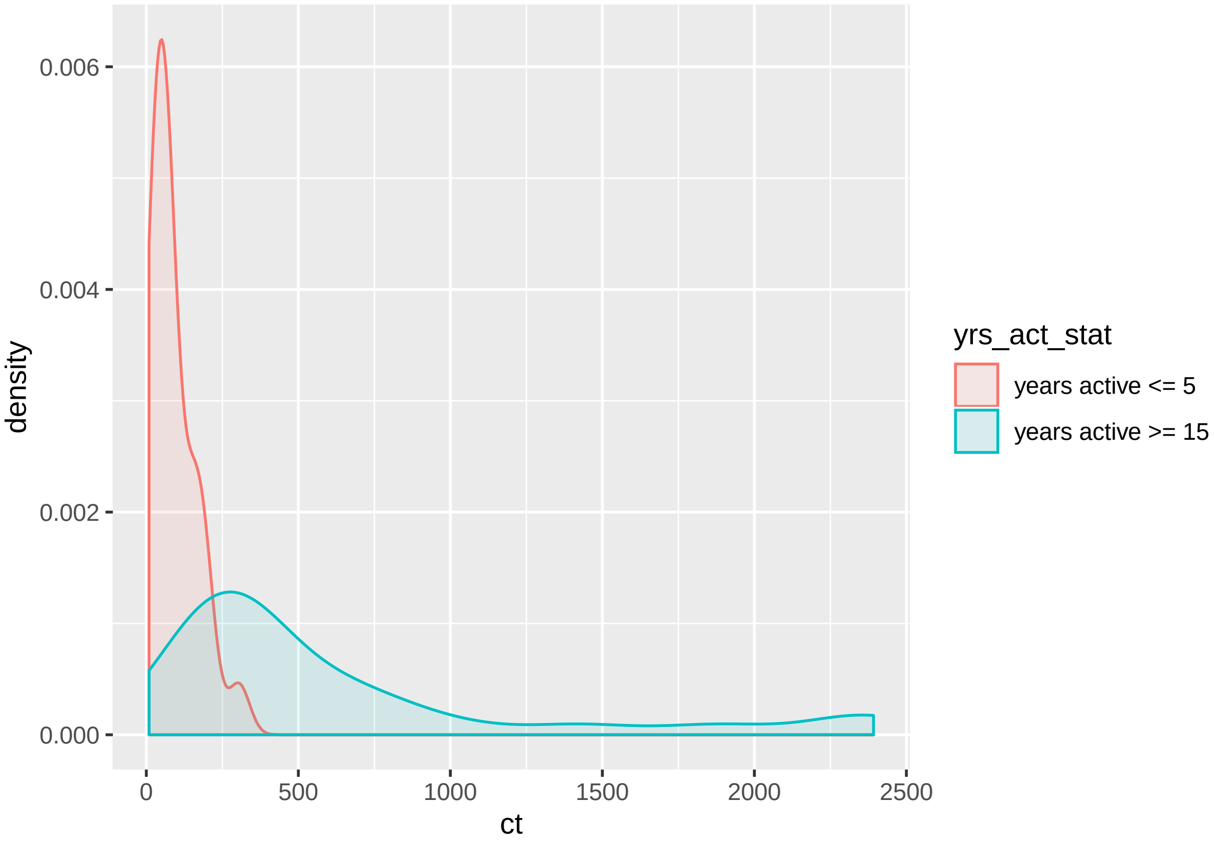
The data is not specific based on either test.

Next I will look at the bout count by community

```
shapiro.test(groups %>% filter(yrs_active >= 15) %>% pull(ct))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  groups %>% filter(yrs_active >= 15) %>% pull(ct)
## W = 0.71955, p-value = 1.351e-05
```

```
ggplot(groups,aes(ct, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0.1)
```



The data is not normally distributed. The data does show some outliers. For safety I will run both tests again.

```
fligner.test(ct ~ yrs_act_stat, groups)
```

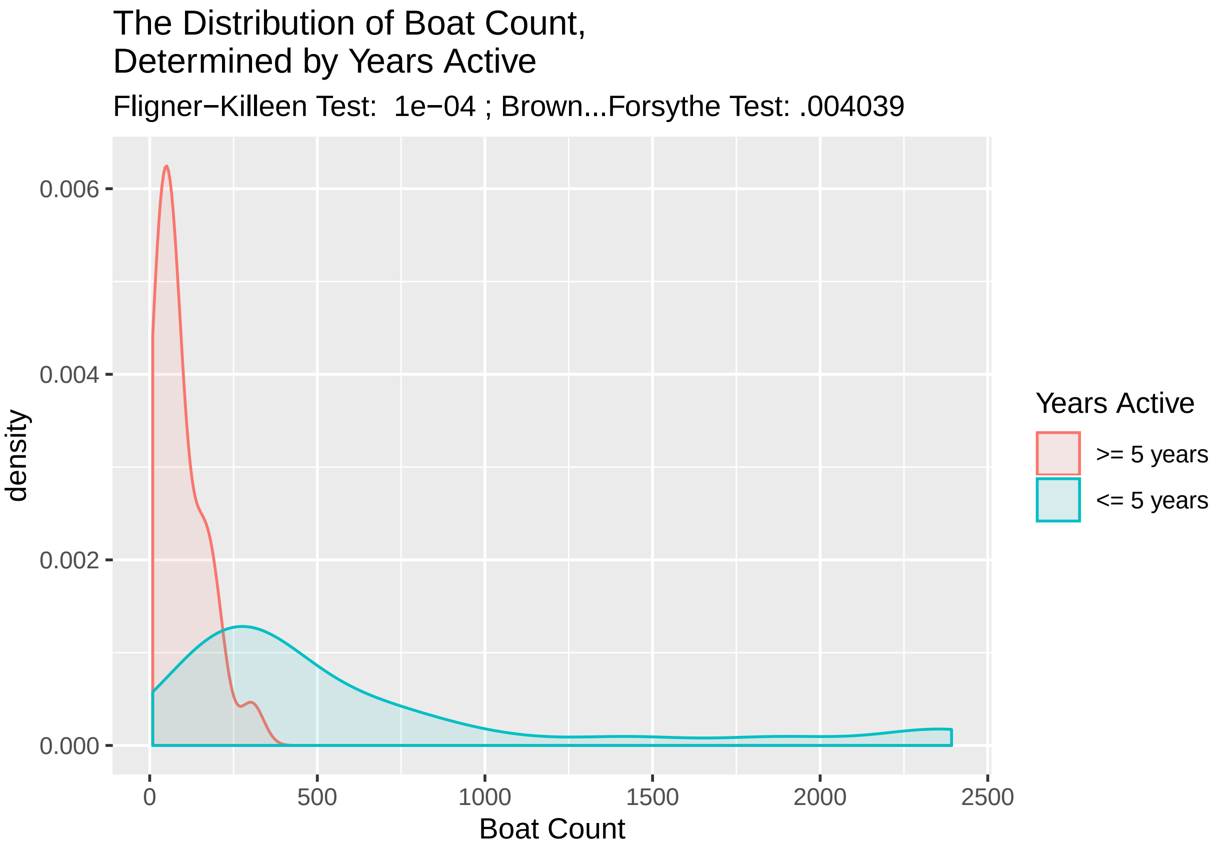
```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  ct by yrs_act_stat
## Fligner-Killeen:med chi-squared = 16.166, df = 1, p-value = 5.803e-05
```

```
leveneTest(ct ~ yrs_act_stat, groups, center = median)
```

```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.

## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value   Pr(>F)
## group  1   9.122 0.004039 **
##       48
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ggplot(groups,aes(ct, fill = yrs_act_stat, colour = yrs_act_stat)) + geom_density(alpha = 0.1) + scale_
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <e2>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <80>

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Fligner-Killeen Test: 1e-04 ; Brown-Forsythe Test: .
## 004039' in 'mbcsToSbcs': dot substituted for <93>
```



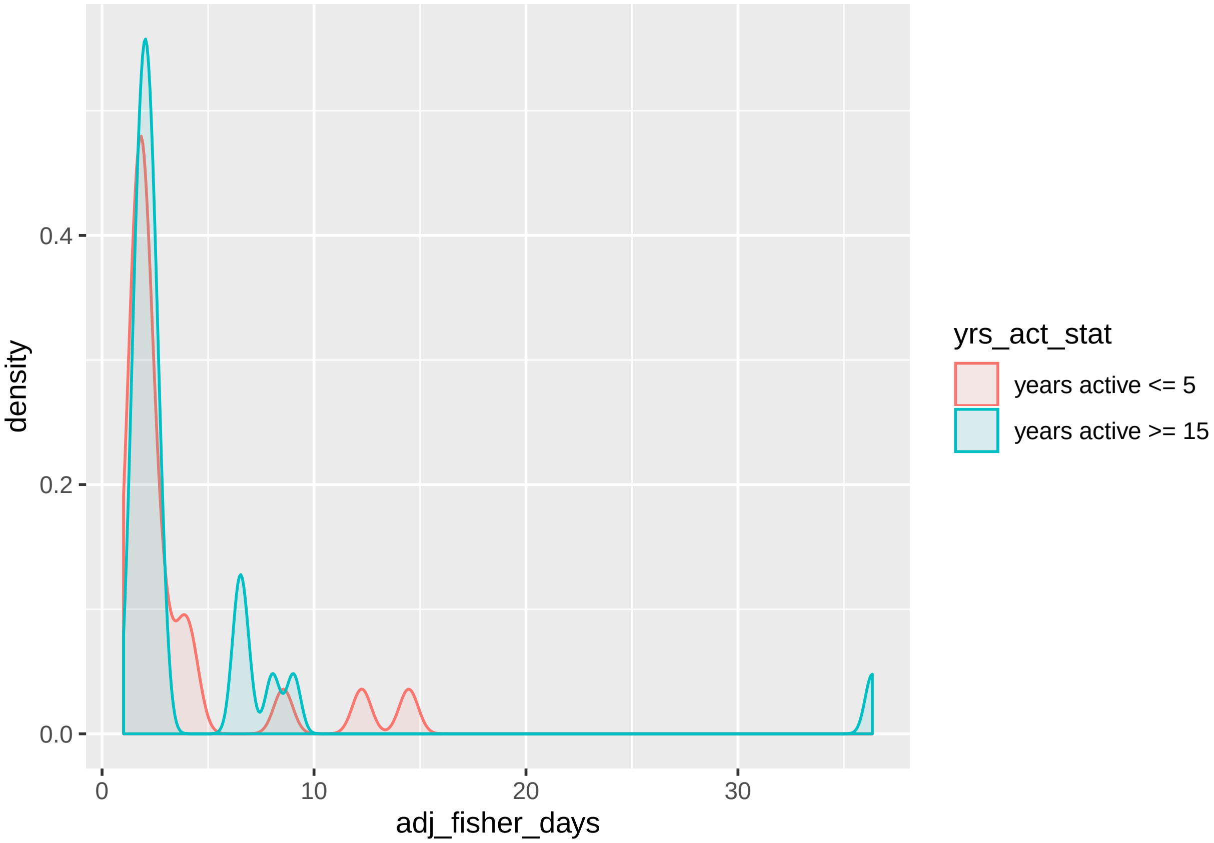
Both tests came back statistically significant regarding the p-values. This indicates the varainces are not equal.

I will now look at the adj_fisher_days

```
shapiro.test(groups %>% filter(yrs_active <= 5) %>% pull(adj_fisher_days))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  groups %>% filter(yrs_active <= 5) %>% pull(adj_fisher_days)
## W = 0.60204, p-value = 4.493e-07
```

```
ggplot(groups,aes(adj_fisher_days, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0
```



This is showing a lot of outliers but again I will use both tests. I will weight the Fligner-Killeen test higher.

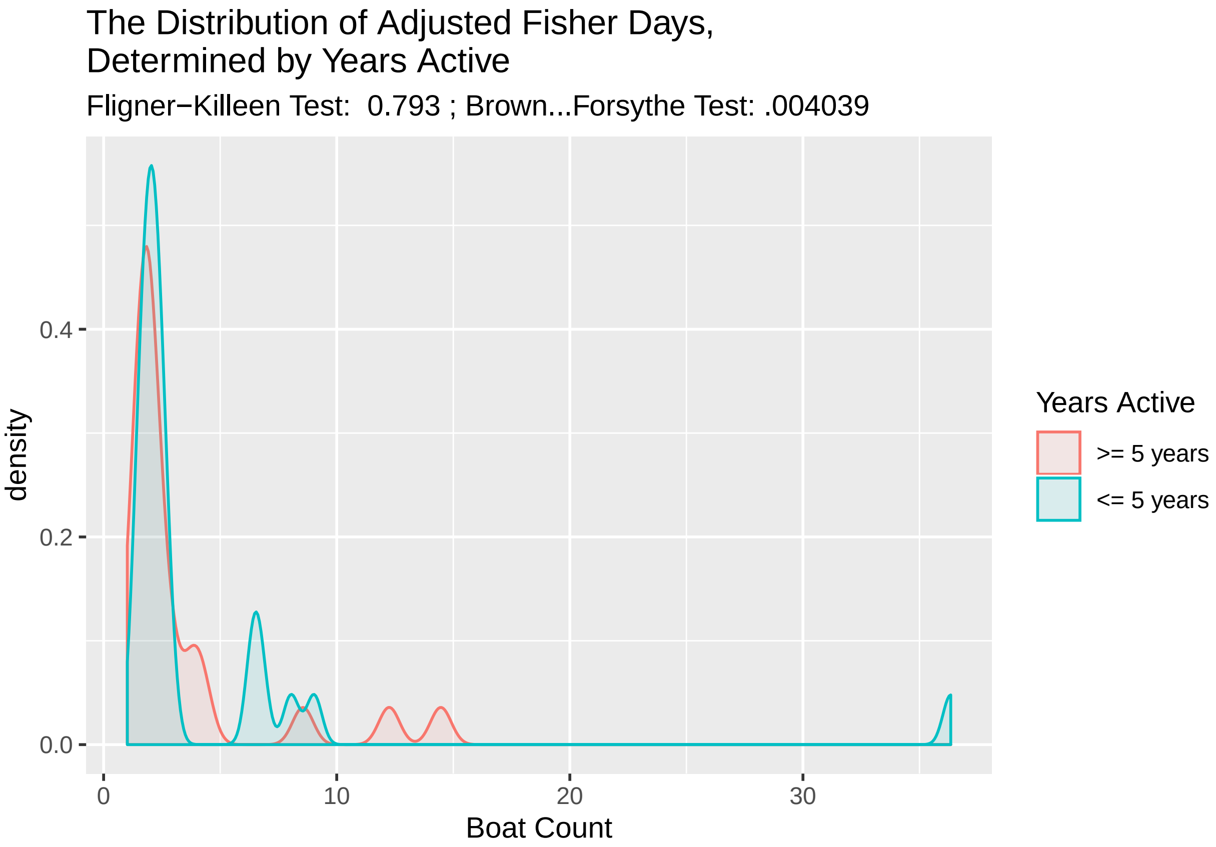
```
fligner.test(adj_fisher_days ~ yrs_act_stat, groups)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  adj_fisher_days by yrs_act_stat
## Fligner-Killeen:med chi-squared = 0.068891, df = 1, p-value = 0.793
```

```
leveneTest(adj_fisher_days ~ yrs_act_stat, groups, center = median,trim=.1)
```

```
## Levene's Test for Homogeneity of Variance (center = median: 0.1)
##       Df F value Pr(>F)
## group  1  0.4011 0.5295
##       48
```

```r
ggplot(groups,aes(adj_fisher_days, fill = yrs_act_stat, colour = yrs_act_stat)) + geom_density(alpha = 
```



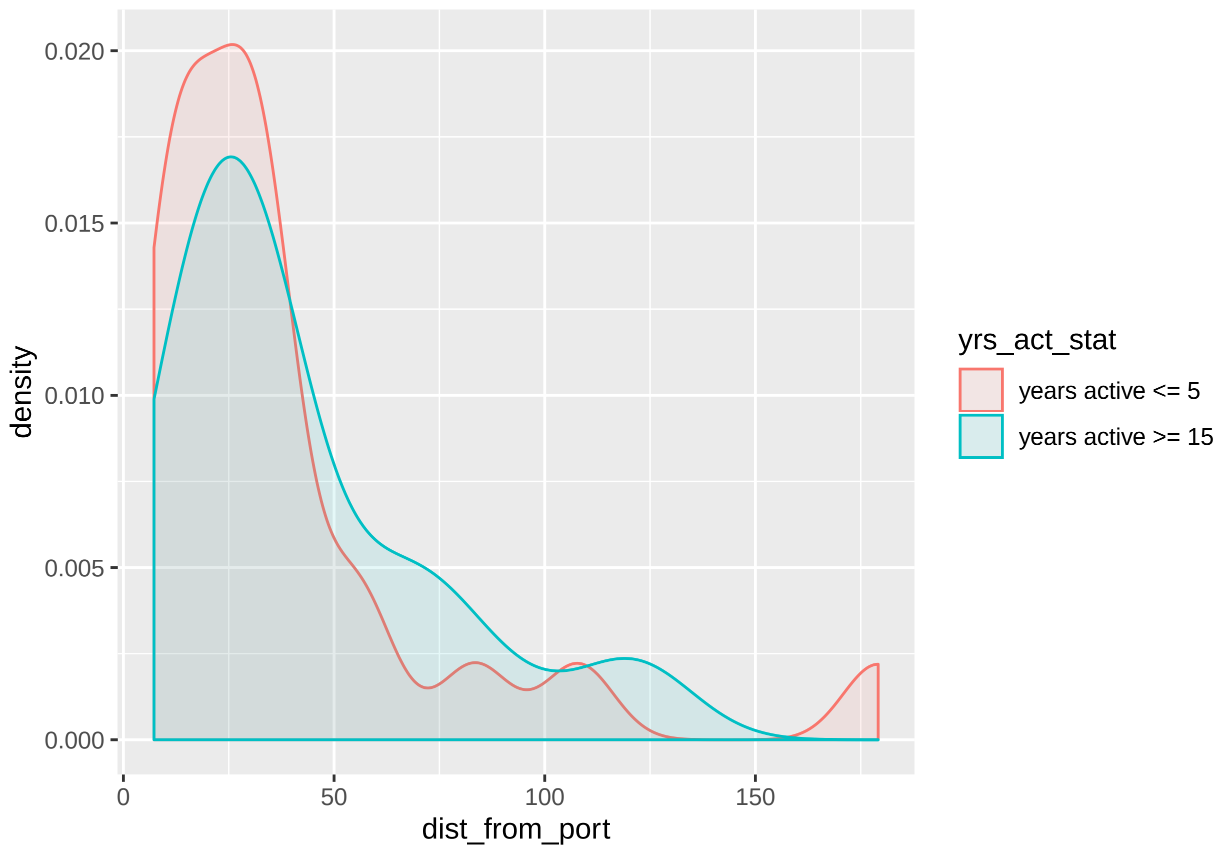
The tests show that the varaince is not significantly different.

Now on to the dist_from_port

```r
shapiro.test(groups %>% filter(yrs_active <= 5) %>% pull(dist_from_port))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  groups %>% filter(yrs_active <= 5) %>% pull(dist_from_port)
## W = 0.70926, p-value = 2.573e-05
```

```r
ggplot(groups,aes(dist_from_port, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0.
```



The data appears to have some issues but doesn't have an extensive amount of problems. Thus I will focus on the Levene test.

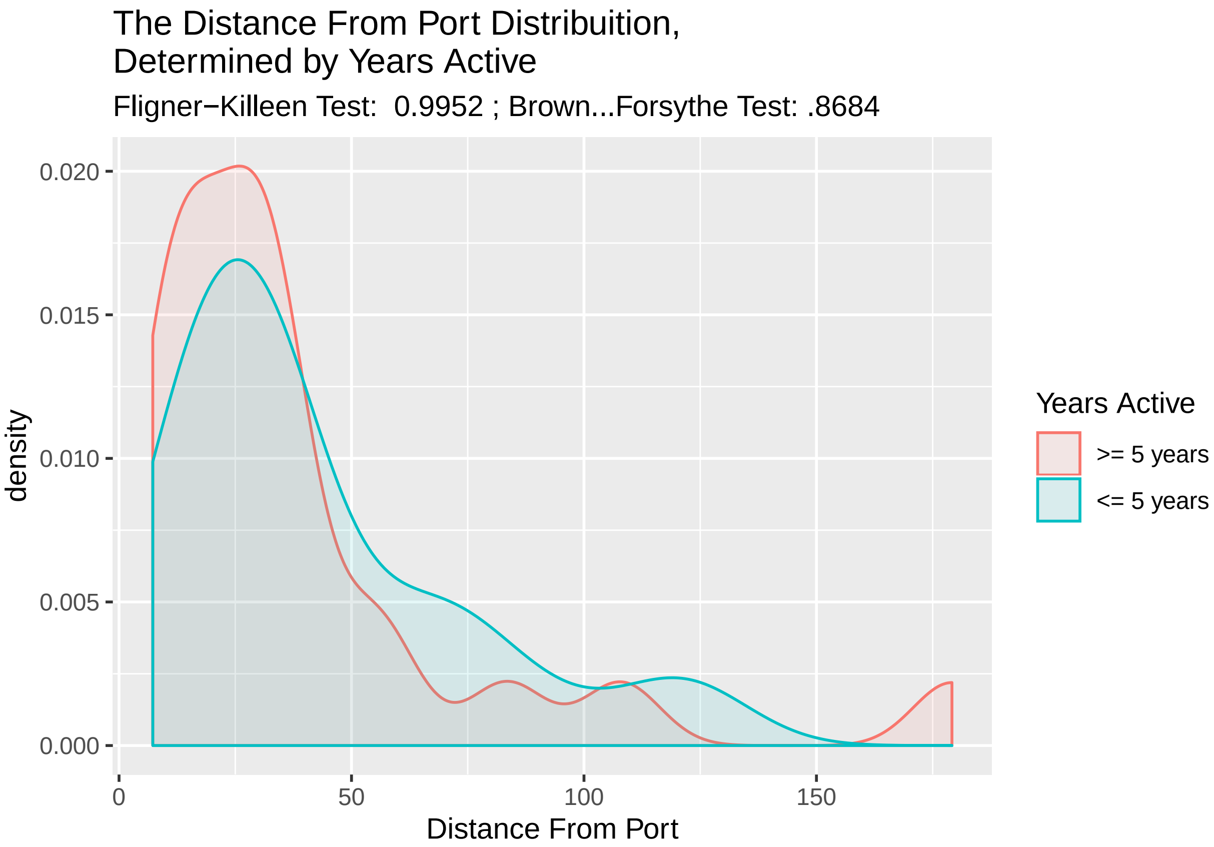
```
fligner.test(dist_from_port ~ yrs_act_stat, groups)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  dist_from_port by yrs_act_stat
## Fligner-Killeen:med chi-squared = 3.686e-05, df = 1, p-value = 0.9952
```

```
leveneTest(dist_from_port ~ yrs_act_stat, groups, center = median)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.0278 0.8684
##       45
```

```
ggplot(groups,aes(dist_from_port, fill = yrs_act_stat, colour = yrs_act_stat)) + geom_density(alpha = 0
```



Based on the tests, the varaince is not signifcantly different.

Now to look at richness

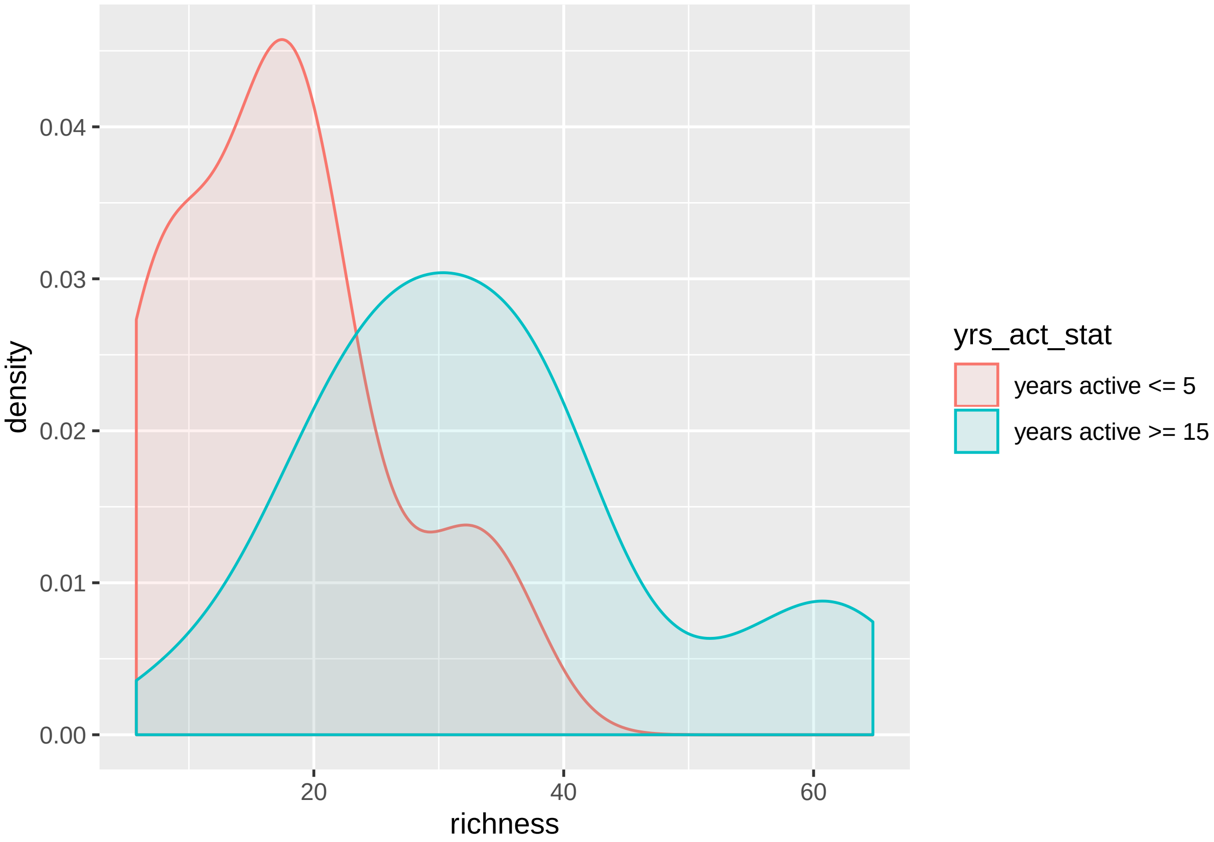
```
shapiro.test(groups %>% filter(yrs_active <= 5) %>% pull(richness))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  groups %>% filter(yrs_active <= 5) %>% pull(richness)
## W = 0.92142, p-value = 0.05519
```

```
shapiro.test(groups %>% filter(yrs_active >= 15) %>% pull(richness))
```

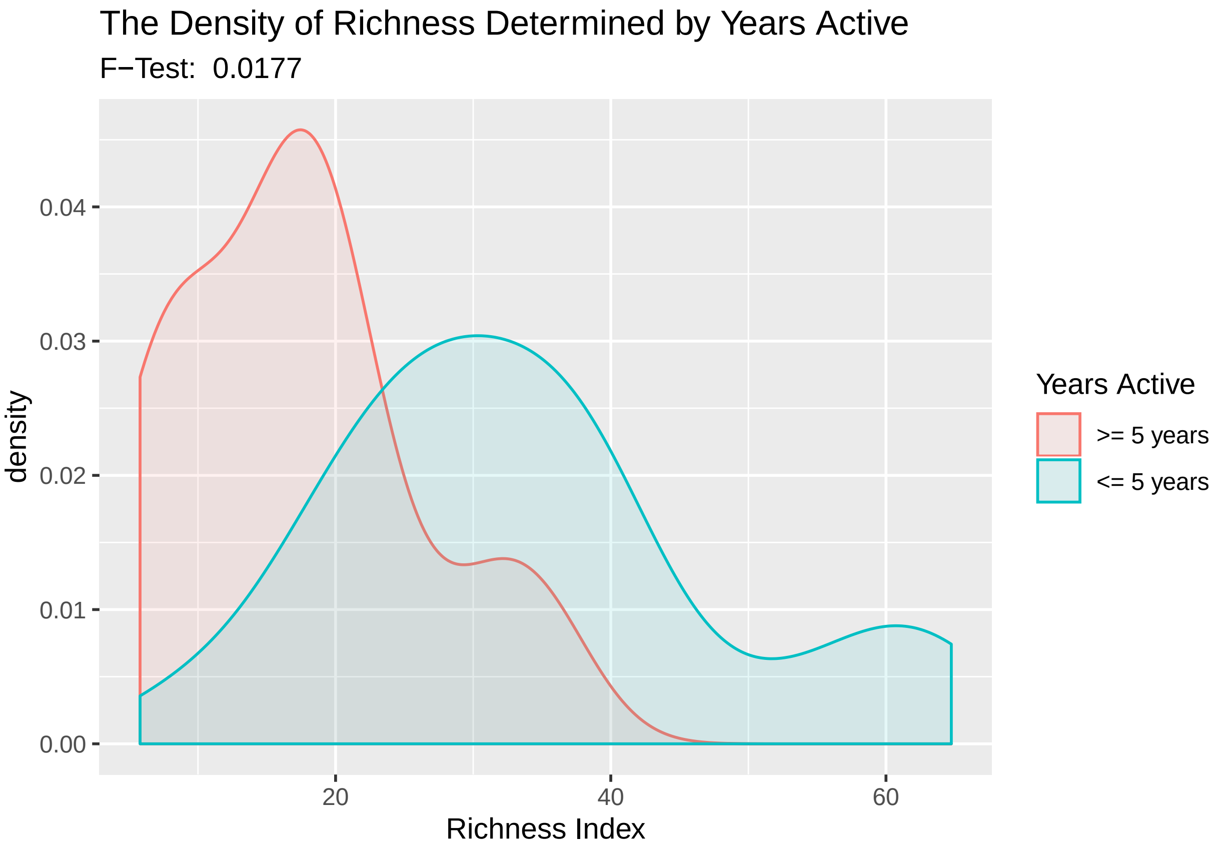
```
##
##  Shapiro-Wilk normality test
##
## data:  groups %>% filter(yrs_active >= 15) %>% pull(richness)
## W = 0.94616, p-value = 0.2051
```

```
ggplot(groups,aes(richness, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0.1)
```



This data is actually showing normality so it is possible to use an F-test.

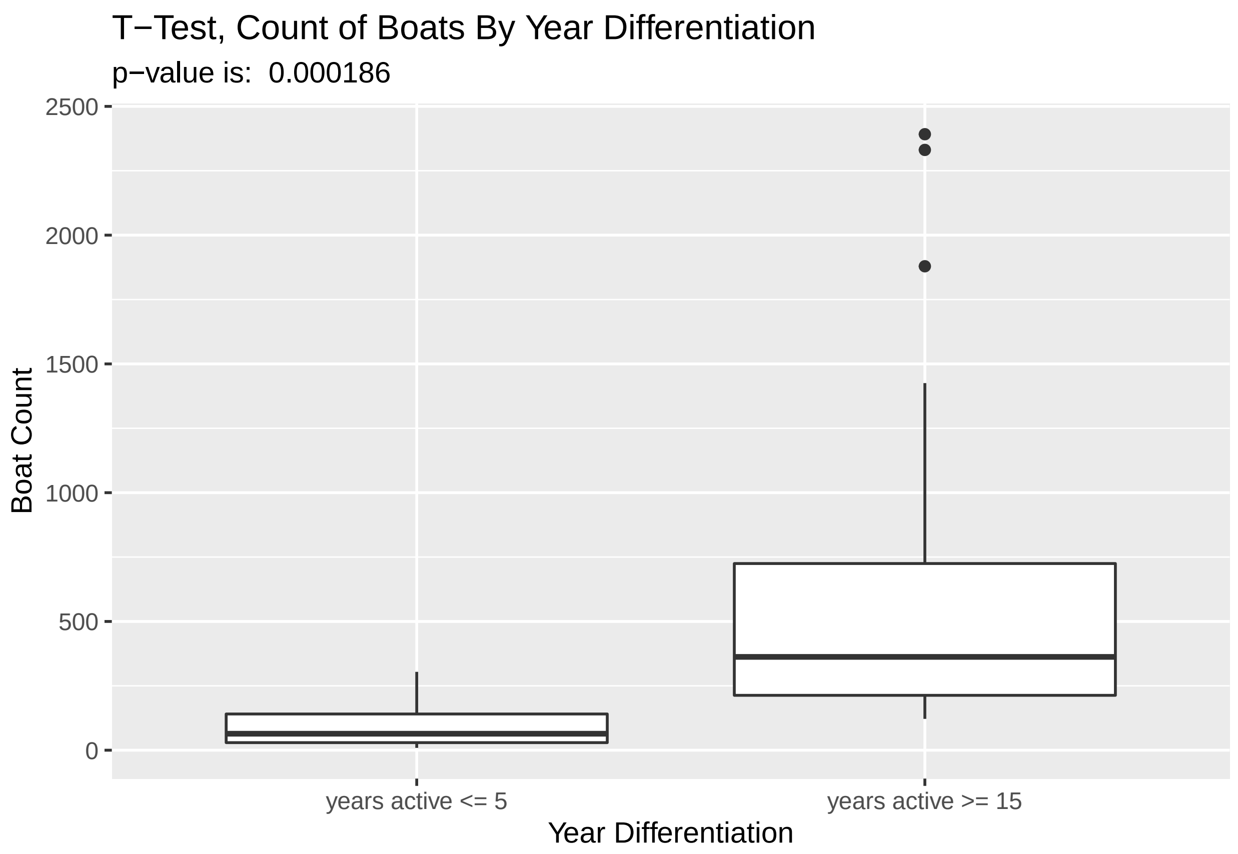
```
ggplot(groups,aes(richness, fill = yrs_act_stat,colour = yrs_act_stat)) + geom_density(alpha = 0.1) + se
```



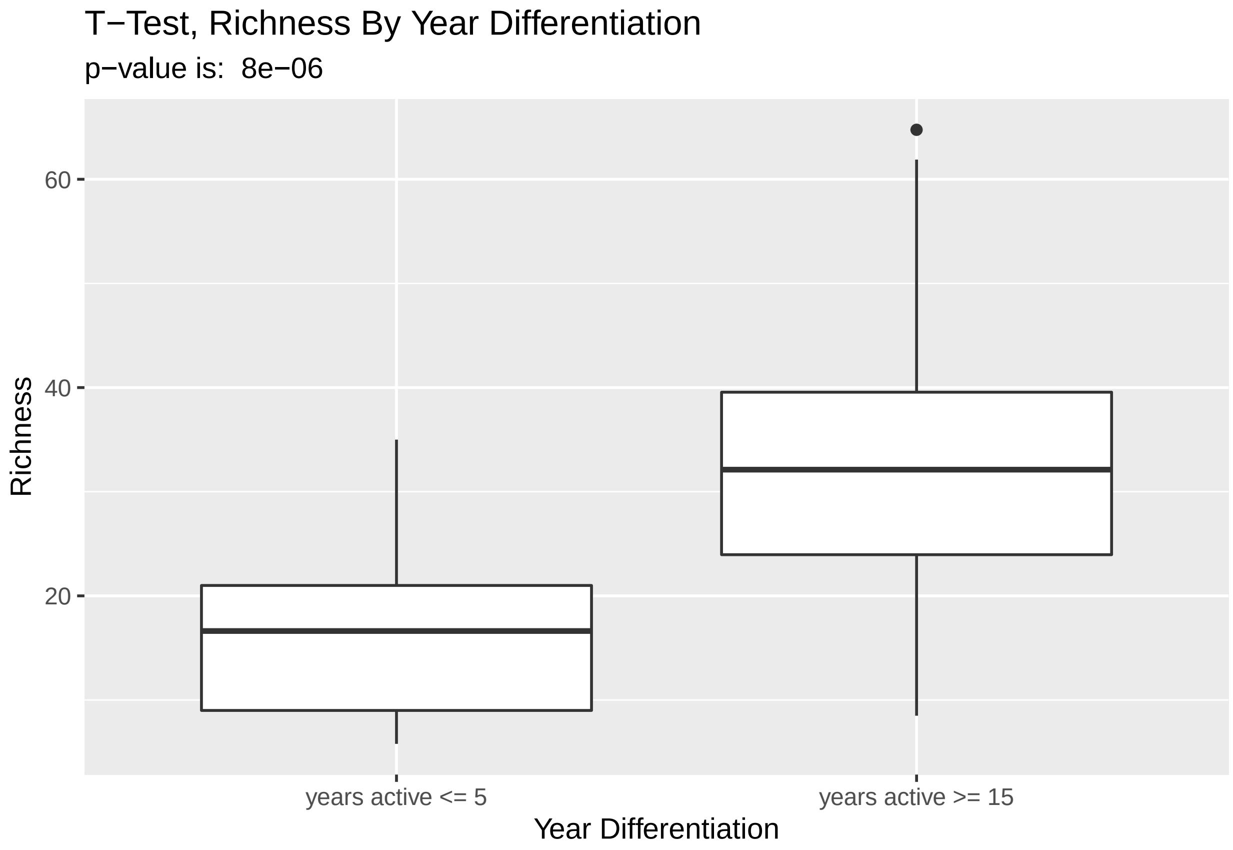
The data is significant at the 5% level. the varainces are not from the same distributions.

I will now compare the means using a T-test

```
ggplot(groups, aes(x=yrs_act_stat,y = ct)) + geom_boxplot() + labs(title="T-Test, Count of Boats By Yea
```



```
ggplot(groups, aes(x=yrs_act_stat,y = richness)) + geom_boxplot() + labs(title="T-Test, Richness By Yea
```


T−Test, Richness By Year Differentiation
p−value is: 8e−06
Richness
60
40
20
years active <= 5
years active >= 15
Year Differentiation