Vulnerability

Laura

3/2/2021

##Calculating perceptions of vulnerability based upon the exposure, sensitivity, and adaptive capacity of survey participants. ##saves figures into folder ("../figures/vulnerability)

#load data

##functions to get likert answers to a 0 to 1 scale

##exposure ##changing from likert responses to numbers in order to calculate exposure index ##modify for different fisheries depending on survey region

##Exposure based on perceptions of how will climate affect all fisheries

##Exposure based upon perceptions of how climate will affect only the fisheries you participate in ##calling other Rmd file with code that calculates exposure that way and adds it to responses as a column called fishery\_specific\_exposure

##sensitivity index

##adaptive capacity

##individual vulnerability

Looking at the correlation between risk and vulnerability ranks when calculated following Cinner (2012), where r = e + s and v = e + s - ac versus using the euclidean distance method (Levin and Samhouri 2012) risk = sqrt(e^2 + s^2) and v = sqrt(e^2 + s^2 (1-ac)^2).

## risk\_rank risk\_euc\_rank vuln\_rank vuln\_euc\_rank  
## risk\_rank 1.0000000 0.9795007 0.9280633 0.9115440  
## risk\_euc\_rank 0.9795007 1.0000000 0.9095548 0.9195278  
## vuln\_rank 0.9280633 0.9095548 1.0000000 0.9849656  
## vuln\_euc\_rank 0.9115440 0.9195278 0.9849656 1.0000000

#summary of vulnerability and its components for all respondents

vulnsummary = responses %>%  
 dplyr::summarise(exposure\_avg = mean(indv\_exposure, na.rm = TRUE),  
 ex\_sd = sd(indv\_exposure, na.rm = TRUE),  
 sensitivity\_avg = mean(indv\_sensitivity, na.rm = TRUE),  
 sensitivity\_sd = sd(indv\_sensitivity, na.rm = TRUE),  
 adaptive\_avg = mean(indv\_ac),  
 adaptive\_sd = sd(indv\_ac),  
 vuln\_euc\_avg = mean(indv\_vulnerability\_euc, na.rm = TRUE),  
 vuln\_euc\_sd = sd(indv\_vulnerability\_euc, na.rm = TRUE),  
 count = n()  
)  
vulnsummary

## exposure\_avg ex\_sd sensitivity\_avg sensitivity\_sd adaptive\_avg adaptive\_sd  
## 1 0.6445076 0.219169 0.5613757 0.2113982 0.3925926 0.1603165  
## vuln\_euc\_avg vuln\_euc\_sd count  
## 1 1.085005 0.2260321 105

Components of vulnerability when we subset by each unique combination of fisheries. #subseting by fishery

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 67 x 7  
## listoffisheries exposure\_avg sensitivity\_avg adaptive\_avg risk\_avg  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 "" NaN 0.361 0.444 NaN   
## 2 "crab" 0.75 0.264 0.389 0.810  
## 3 "geoduck or ho~ 0.625 0.306 0.444 0.741  
## 4 "geoduck or ho~ 0.583 0.653 0.417 0.876  
## 5 "groundfish" 0.875 0.722 0.208 1.16   
## 6 "groundfish, h~ 0.4 0.278 0.389 0.487  
## 7 "halibut longl~ 0.5 0.491 0.574 NaN   
## 8 "halibut longl~ 0.5 0.25 0.639 0.559  
## 9 "halibut longl~ 1 0.5 0.222 1.12   
## 10 "halibut longl~ 0.5 0.556 0.417 0.747  
## # ... with 57 more rows, and 2 more variables: vuln\_euc\_avg <dbl>, count <int>

Or we can look at vulnerability based upon everyone that participates in a particular fishery regardless of what else they fish for.

##average vulnerability by fishery

## fishery avg\_exposure avg\_sensitivity avg\_ac ac\_sd  
## 1 salmon troll 0.6487981 0.5843621 0.3878601 0.20067572  
## 2 salmon seine 0.7240385 0.5470085 0.3888889 0.14027090  
## 3 salmon gillnet 0.6393018 0.5884503 0.3764620 0.15374013  
## 4 herring roe gillnet 0.6080208 0.5708333 0.4083333 0.14281014  
## 5 herring roe seine 0.6473485 0.6439394 0.4267677 0.15329344  
## 6 herring spawn on kelp 0.6116667 0.6611111 0.2944444 0.08914893  
## 7 tuna troll 0.5769676 0.6203704 0.4290123 0.14519108  
## 8 tuna international 0.6423611 0.4791667 0.4791667 0.09453971  
## 9 tuna US 0.5763889 0.6759259 0.6018519 0.05782406  
## 10 hake 0.7812500 0.6111111 0.4166667 0.23570226  
## 11 sardine 0.6166667 0.6481481 0.3703704 0.13127266  
## 12 groundfish 0.7072917 0.5648148 0.3333333 0.17391640  
## 13 halibut longline 0.5697173 0.5157407 0.4648148 0.14087343  
## 14 sablefish longline 0.6337500 0.5222222 0.4638889 0.14875455  
## 15 sablefish trap 0.5859375 0.7152778 0.5000000 0.12213802  
## 16 rockfish 0.5915278 0.4629630 0.4944444 0.15294003  
## 17 lingcod 0.6202546 0.5409357 0.4005848 0.15830782  
## 18 dogfish 0.7300000 0.4444444 0.4500000 0.17055646  
## 19 shrimp trawl 0.5000000 0.5694444 0.3333333 0.07856742  
## 20 euphausiid 0.2083333 0.8333333 0.5833333 0.07856742  
## 21 prawn shrimp trap 0.5640351 0.5716374 0.4195906 0.13253838  
## 22 crab 0.6229167 0.4131944 0.4201389 0.15173708  
## 23 geoduck or horseclam 0.6333333 0.4500000 0.4166667 0.03402069  
## 24 red urchin 0.5440476 0.7142857 0.3968254 0.15105449  
## 25 green urchin 0.5729167 0.6805556 0.3750000 0.12318643  
## 26 sea cucumber 0.6236111 0.6712963 0.3981481 0.10343882  
## avg\_risk risk\_sd avg\_risk\_euc avg\_vulnerability avg\_vulnerabilty\_euc  
## 1 1.2091307 0.37205399 0.9009308 0.8212706 1.0968241  
## 2 1.2710470 0.23110839 0.9285212 0.8821581 1.1241473  
## 3 1.2109284 0.32351825 0.8990224 0.8344664 1.1035129  
## 4 1.1788542 0.34075188 0.8548285 0.7705208 1.0493256  
## 5 1.2912879 0.23968866 0.9359068 0.8645202 1.1120159  
## 6 1.2727778 0.21367594 0.9085747 0.9783333 1.1580847  
## 7 1.1973380 0.29126823 0.8609385 0.7683256 1.0429329  
## 8 0.9609375 0.72363074 0.9073614 0.4817708 1.0689902  
## 9 1.2523148 0.31398823 0.8912167 0.6504630 0.9785275  
## 10 1.3923611 0.45667313 0.9925224 0.9756944 1.1800542  
## 11 1.2648148 0.19636617 0.9024180 0.8944444 1.1102697  
## 12 1.2721065 0.42417095 0.9133000 0.9387731 1.1495696  
## 13 1.0474769 0.32817998 0.7897275 0.5826620 0.9691596  
## 14 1.1559722 0.39131631 0.8369155 0.6920833 1.0099499  
## 15 1.3012153 0.33887389 0.9385892 0.8012153 1.0751457  
## 16 1.0544907 0.29428985 0.7679961 0.5600463 0.9310340  
## 17 1.1285453 0.39289035 0.8470840 0.7279605 1.0436144  
## 18 1.1744444 0.25608641 0.8701212 0.7244444 1.0317256  
## 19 1.0694444 0.09820928 0.7591946 0.7361111 1.0132197  
## 20 1.0416667 0.45176267 0.8772473 0.4583333 0.9782568  
## 21 1.1356725 0.19360104 0.8300547 0.7160819 1.0214091  
## 22 1.0361111 0.18021641 0.7793571 0.6159722 0.9746099  
## 23 1.0833333 0.19934306 0.8107082 0.6666667 0.9999471  
## 24 1.2583333 0.17583575 0.9084512 0.8615079 1.1001892  
## 25 1.2534722 0.18988538 0.8967701 0.8784722 1.0947374  
## 26 1.2949074 0.16082750 0.9234263 0.8967593 1.1044422  
## vuln\_sd ex\_sd sen\_sd  
## 1 0.272407596 0.21186848 0.22494083  
## 2 0.122210220 0.18719229 0.18051446  
## 3 0.208889415 0.24271909 0.21456936  
## 4 0.232544135 0.18929486 0.23715798  
## 5 0.125242787 0.19520531 0.18914125  
## 6 0.091002932 0.18386287 0.08193267  
## 7 0.197283436 0.12957020 0.21809900  
## 8 0.250658410 0.16612325 0.37773352  
## 9 0.223296322 0.08419691 0.23295825  
## 10 0.151841143 0.22097087 0.23570226  
## 11 0.032076565 0.18929694 0.04243126  
## 12 0.267987037 0.23353321 0.23744178  
## 13 0.200242259 0.16170914 0.22527635  
## 14 0.251195326 0.20221049 0.25452018  
## 15 0.205031670 0.17644390 0.25294053  
## 16 0.213422964 0.16837014 0.21015517  
## 17 0.260754009 0.22002697 0.21182636  
## 18 0.250799059 0.22803509 0.14566790  
## 19 0.003500175 0.00000000 0.09820928  
## 20 0.163137199 0.29462783 0.15713484  
## 21 0.137956376 0.17976435 0.18456279  
## 22 0.206423964 0.20566144 0.19999862  
## 23 0.092843559 0.12638126 0.24798571  
## 24 0.115510970 0.15772649 0.11307783  
## 25 0.160764159 0.16796370 0.06612434  
## 26 0.126187569 0.15279040 0.05936045

Components of vulnerability based upon the unique combination of regions someone fishes in.

#use the same method to look at regions #subseting by region

## `summarise()` ungrouping output (override with `.groups` argument)

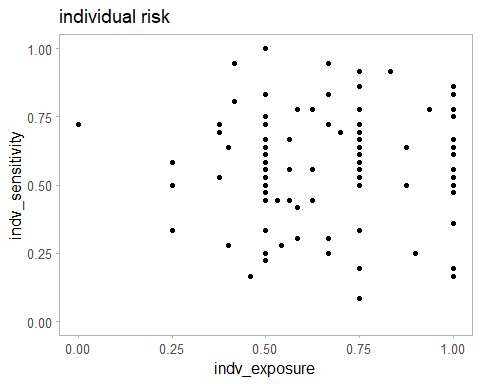
## # A tibble: 50 x 7  
## listofregions exposure\_avg sensitivity\_avg adaptive\_avg risk\_avg vuln\_euc\_avg  
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 central coast 0.562 0.611 0.389 0.859 1.06   
## 2 Haida Gwaii 0.672 0.465 0.514 0.819 0.955  
## 3 Haida Gwaii,~ 0.5 0.25 0.639 0.559 0.666  
## 4 Haida Gwaii,~ 0.25 0.639 0.472 0.735 0.906  
## 5 Haida Gwaii,~ 0.75 0.509 0.537 NaN 1.05   
## 6 Haida Gwaii,~ 0.646 0.676 0.398 0.940 1.12   
## 7 Haida Gwaii,~ 0.558 0.583 0.306 0.829 1.08   
## 8 Haida Gwaii,~ 0.5 0.556 0.25 0.747 1.06   
## 9 Haida Gwaii,~ 0.521 0.514 0.264 0.755 1.06   
## 10 Haida Gwaii,~ 0.725 0.389 0.431 0.870 1.04   
## # ... with 40 more rows, and 1 more variable: count <int>

Or if we average by anyone that fishes in a particular region. ##average vulnerability by region

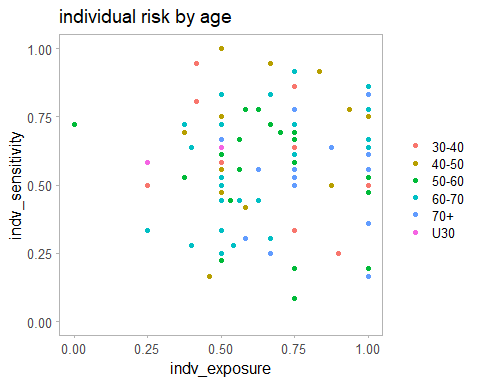
## region avg\_exposure avg\_sensitivity avg\_ac avg\_risk  
## 1 offshore 0.5562500 0.5717593 0.5092593 1.0816551  
## 2 Haida Gwaii 0.6079776 0.5334596 0.4324495 1.0999842  
## 3 north coast 0.6252558 0.5550926 0.3773148 1.1490856  
## 4 central coast 0.6146850 0.5535714 0.3935185 1.1536210  
## 5 NVI 0.6556052 0.5598765 0.3697531 1.1717747  
## 6 WCVI 0.6668356 0.5935673 0.4027778 1.2428545  
## 7 SG 0.6768519 0.5555556 0.3900966 1.2176932  
## 8 SJF 0.6527778 0.5254630 0.4305556 1.1782407  
## 9 IW 0.6979167 0.4907407 0.5185185 0.9560185  
## 10 USW 0.5322917 0.5763889 0.5486111 1.1086806  
## avg\_vulnerability avg\_risk\_euc avg\_vulnerability\_euc  
## 1 0.5723958 0.8522650 0.9990761  
## 2 0.6675347 0.8426071 1.0255192  
## 3 0.7717708 0.8685346 1.0809517  
## 4 0.7601025 0.8577152 1.0632117  
## 5 0.8020216 0.8861715 1.0944004  
## 6 0.8400768 0.9170305 1.1068825  
## 7 0.8275966 0.9051877 1.0963290  
## 8 0.7476852 0.8600172 1.0383122  
## 9 0.4375000 1.0147195 1.1323335  
## 10 0.5600694 0.7901604 0.9292506

##exploratory figures

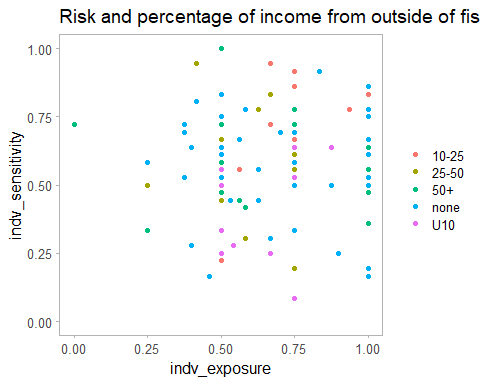
Visualizing risk.

##visualizing individual risk (exposure v sensitivity) 

## Saving 5 x 4 in image



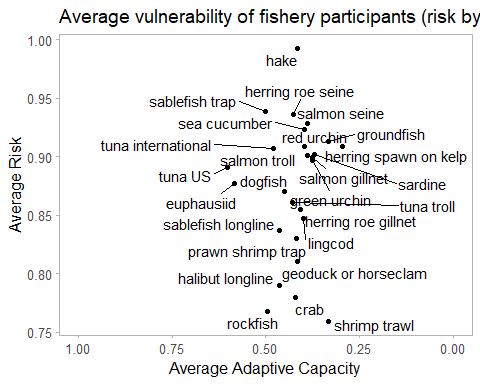
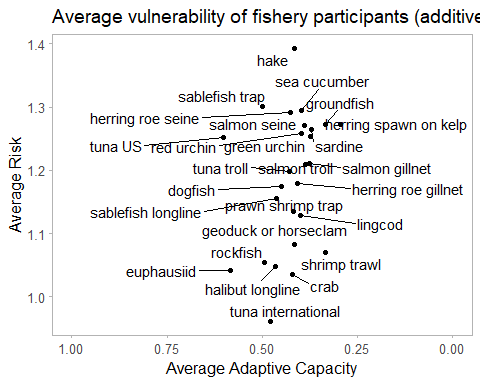
## Saving 5 x 4 in image



## Saving 5 x 4 in image

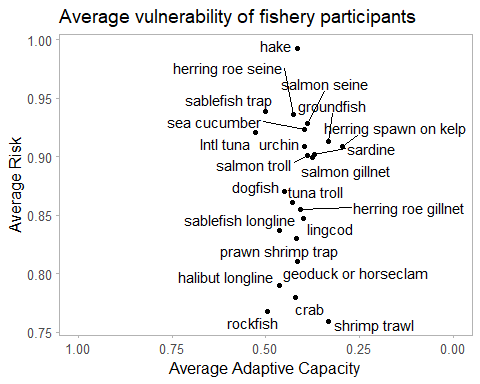
First plot is additive risk, second is euclidean distance. ##considering average vulnerability based upon fishery participation

## Saving 5 x 4 in image  
## Saving 5 x 4 in image

 ##fishery vulnerability with a few consolidations of fisheries

## fishery avg\_exposure avg\_sensitivity avg\_ac ac\_sd  
## 1 salmon troll 0.6487981 0.5843621 0.3878601 0.20067572  
## 2 salmon seine 0.7240385 0.5470085 0.3888889 0.14027090  
## 3 salmon gillnet 0.6393018 0.5884503 0.3764620 0.15374013  
## 4 herring roe gillnet 0.6080208 0.5708333 0.4083333 0.14281014  
## 5 herring roe seine 0.6473485 0.6439394 0.4267677 0.15329344  
## 6 herring spawn on kelp 0.6116667 0.6611111 0.2944444 0.08914893  
## 7 tuna troll 0.5769676 0.6203704 0.4290123 0.14519108  
## 10 hake 0.7812500 0.6111111 0.4166667 0.23570226  
## 11 sardine 0.6166667 0.6481481 0.3703704 0.13127266  
## 12 groundfish 0.7072917 0.5648148 0.3333333 0.17391640  
## 13 halibut longline 0.5697173 0.5157407 0.4648148 0.14087343  
## 14 sablefish longline 0.6337500 0.5222222 0.4638889 0.14875455  
## 15 sablefish trap 0.5859375 0.7152778 0.5000000 0.12213802  
## 16 rockfish 0.5915278 0.4629630 0.4944444 0.15294003  
## 17 lingcod 0.6202546 0.5409357 0.4005848 0.15830782  
## 18 dogfish 0.7300000 0.4444444 0.4500000 0.17055646  
## 19 shrimp trawl 0.5000000 0.5694444 0.3333333 0.07856742  
## 21 prawn shrimp trap 0.5640351 0.5716374 0.4195906 0.13253838  
## 22 crab 0.6229167 0.4131944 0.4201389 0.15173708  
## 23 geoduck or horseclam 0.6333333 0.4500000 0.4166667 0.03402069  
## 26 sea cucumber 0.6236111 0.6712963 0.3981481 0.10343882  
## 27 Intl tuna 0.6187500 0.5648148 0.5277778 0.10829771  
## 28 urchin 0.5440476 0.7142857 0.3968254 0.15105449  
## avg\_risk risk\_sd avg\_risk\_euc avg\_vulnerability avg\_vulnerabilty\_euc  
## 1 1.209131 0.37205399 0.9009308 0.8212706 1.0968241  
## 2 1.271047 0.23110839 0.9285212 0.8821581 1.1241473  
## 3 1.210928 0.32351825 0.8990224 0.8344664 1.1035129  
## 4 1.178854 0.34075188 0.8548285 0.7705208 1.0493256  
## 5 1.291288 0.23968866 0.9359068 0.8645202 1.1120159  
## 6 1.272778 0.21367594 0.9085747 0.9783333 1.1580847  
## 7 1.197338 0.29126823 0.8609385 0.7683256 1.0429329  
## 10 1.392361 0.45667313 0.9925224 0.9756944 1.1800542  
## 11 1.264815 0.19636617 0.9024180 0.8944444 1.1102697  
## 12 1.272106 0.42417095 0.9133000 0.9387731 1.1495696  
## 13 1.047477 0.32817998 0.7897275 0.5826620 0.9691596  
## 14 1.155972 0.39131631 0.8369155 0.6920833 1.0099499  
## 15 1.301215 0.33887389 0.9385892 0.8012153 1.0751457  
## 16 1.054491 0.29428985 0.7679961 0.5600463 0.9310340  
## 17 1.128545 0.39289035 0.8470840 0.7279605 1.0436144  
## 18 1.174444 0.25608641 0.8701212 0.7244444 1.0317256  
## 19 1.069444 0.09820928 0.7591946 0.7361111 1.0132197  
## 21 1.135673 0.19360104 0.8300547 0.7160819 1.0214091  
## 22 1.036111 0.18021641 0.7793571 0.6159722 0.9746099  
## 23 1.083333 0.19934306 0.8107082 0.6666667 0.9999471  
## 26 1.294907 0.16082750 0.9234263 0.8967593 1.1044422  
## 27 1.080440 0.61845499 0.9210269 0.5526620 1.0471183  
## 28 1.258333 0.17583575 0.9084512 0.8615079 1.1001892  
## vuln\_sd ex\_sd sen\_sd  
## 1 0.272407596 0.2118685 0.22494083  
## 2 0.122210220 0.1871923 0.18051446  
## 3 0.208889415 0.2427191 0.21456936  
## 4 0.232544135 0.1892949 0.23715798  
## 5 0.125242787 0.1952053 0.18914125  
## 6 0.091002932 0.1838629 0.08193267  
## 7 0.197283436 0.1295702 0.21809900  
## 10 0.151841143 0.2209709 0.23570226  
## 11 0.032076565 0.1892969 0.04243126  
## 12 0.267987037 0.2335332 0.23744178  
## 13 0.200242259 0.1617091 0.22527635  
## 14 0.251195326 0.2022105 0.25452018  
## 15 0.205031670 0.1764439 0.25294053  
## 16 0.213422964 0.1683701 0.21015517  
## 17 0.260754009 0.2200270 0.21182636  
## 18 0.250799059 0.2280351 0.14566790  
## 19 0.003500175 0.0000000 0.09820928  
## 21 0.137956376 0.1797643 0.18456279  
## 22 0.206423964 0.2056614 0.19999862  
## 23 0.092843559 0.1263813 0.24798571  
## 26 0.126187569 0.1527904 0.05936045  
## 27 0.235205808 0.1353365 0.34724074  
## 28 0.115510970 0.1577265 0.11307783

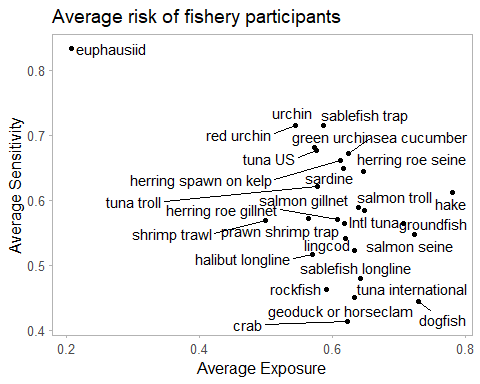
## Saving 5 x 4 in image



##considering average risk based upon fishery participation

## salmon\_troll salmon\_seine salmon\_gillnet   
## 27 13 38   
## herring\_roe\_gillnet herring\_roe\_seine herring\_spawn\_on\_kelp   
## 20 11 5   
## tuna\_troll tuna\_international tuna\_us   
## 18 4 3   
## hake\_midtrawl sardine groundfish\_trawl   
## 2 3 6   
## halibut\_longline lingcod\_hl sablefish\_longline   
## 30 19 10   
## sablefish\_trap rockfish\_hl dogfish\_hl   
## 4 15 5   
## shrimp\_trawl euphausiid\_trawl prawn\_shrimp\_trap   
## 2 2 19   
## crab\_trap geoduck\_horseclam\_dive redseaurchin\_dive   
## 8 5 7   
## greenseaurchin\_dive seacucumber\_dive other   
## 4 6 3

## Saving 5 x 4 in image



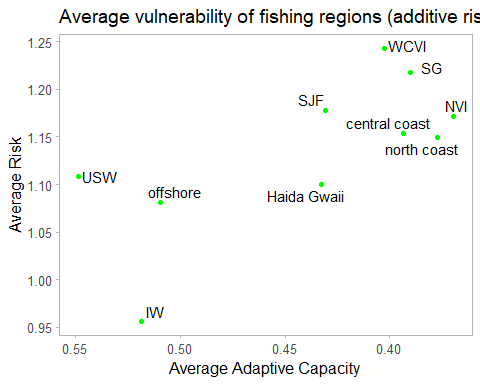
##considering average vulnerability based upon region fished

## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.  
  
## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database

## Warning in grid.Call.graphics(C\_text, as.graphicsAnnot(x$label), x$x, x$y, :  
## font family not found in Windows font database

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database



## Saving 5 x 4 in image

## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.

## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database

## Warning in grid.Call.graphics(C\_text, as.graphicsAnnot(x$label), x$x, x$y, :  
## font family not found in Windows font database

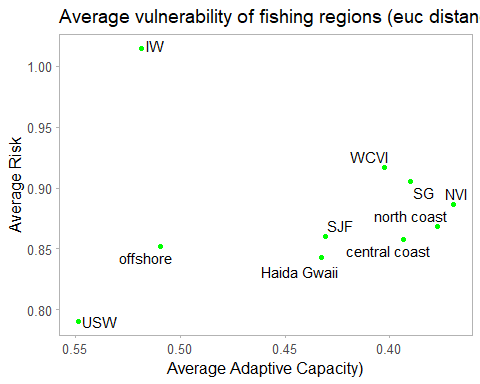
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database

## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.  
  
## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database

## Warning in grid.Call.graphics(C\_text, as.graphicsAnnot(x$label), x$x, x$y, :  
## font family not found in Windows font database

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database



## Saving 5 x 4 in image

## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.

## Warning: Use of `region\_summary$region` is discouraged. Use `region` instead.

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database  
  
## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database

## Warning in grid.Call.graphics(C\_text, as.graphicsAnnot(x$label), x$x, x$y, :  
## font family not found in Windows font database

## Warning in grid.Call(C\_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font  
## family not found in Windows font database

##3d plot

##exploring and visualizing differences in vulnerability and adaptive capacity between groups

##anova of vulnerability for vessel length

#may need to adjust for different surveys  
  
responses$vessel <- factor(responses$vessel, levels = c("U25", "25-35", "36-45", "46-55", "56-65", "66-85", "86+"))  
  
responses %>%  
 group\_by(vessel) %>%  
 dplyr::summarise(mean = mean(indv\_vulnerability),  
 count = n(),  
 sd = sd(indv\_vulnerability)  
 )

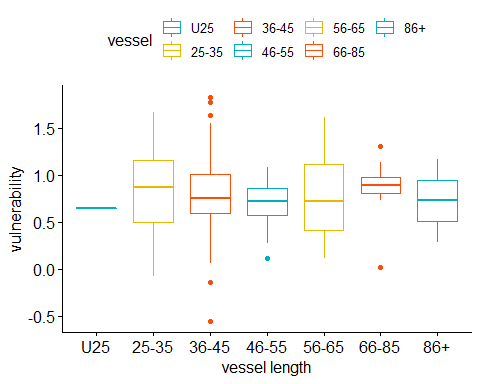
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 7 x 4  
## vessel mean count sd  
## <fct> <dbl> <int> <dbl>  
## 1 U25 0.650 1 NA   
## 2 25-35 0.793 24 0.467  
## 3 36-45 0.782 47 0.477  
## 4 46-55 0.661 9 0.306  
## 5 56-65 0.780 13 0.433  
## 6 66-85 0.841 9 0.357  
## 7 86+ 0.728 2 0.621

# Box plots  
# Plot weight by group and color by group  
  
vessel\_vulnerability<-ggboxplot(responses, x = "vessel", y = "indv\_vulnerability",  
 color = "vessel", palette = c("#00AFBB", "#E7B800", "#FC4E07", "#00AFBB", "#E7B800", "#FC4E07", "#00AFBB"),  
 order = c("U25", "25-35", "36-45", "46-55", "56-65", "66-85", "86+"),  
 ylab = "vulnerability", xlab = "vessel length")  
  
ggsave(plot = vessel\_vulnerability, file = paste0("../figures/vulnerability/vesselvuln.png"))

## Saving 5 x 4 in image

vessel\_vulnerability



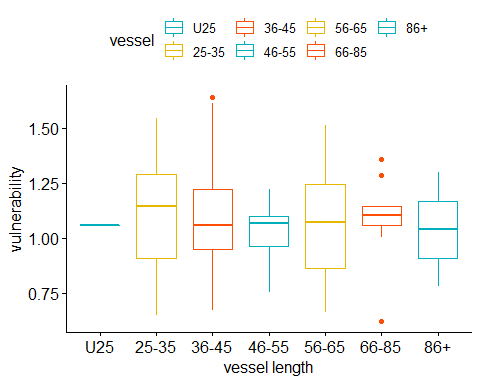
#anova  
vessel\_aov<-aov(indv\_vulnerability ~ vessel, data = responses)  
summary(vessel\_aov)

## Df Sum Sq Mean Sq F value Pr(>F)  
## vessel 6 0.185 0.03087 0.152 0.988  
## Residuals 98 19.892 0.20298

#repeat but use the euclidean distance calc of vulnerabilty  
vessel\_eucvulnerability<-ggboxplot(responses, x = "vessel", y = "indv\_vulnerability\_euc",  
 color = "vessel", palette = c("#00AFBB", "#E7B800", "#FC4E07", "#00AFBB", "#E7B800", "#FC4E07", "#00AFBB"),  
 order = c("U25", "25-35", "36-45", "46-55", "56-65", "66-85", "86+"),  
 ylab = "vulnerability", xlab = "vessel length")  
  
ggsave(plot = vessel\_eucvulnerability, file = paste0("../figures/vulnerability/vesselvuln\_euc.png"))

## Saving 5 x 4 in image

vessel\_eucvulnerability



#anova  
euc\_vessel\_aov<-aov(indv\_vulnerability\_euc ~ vessel, data = responses)  
summary(euc\_vessel\_aov)

## Df Sum Sq Mean Sq F value Pr(>F)  
## vessel 6 0.043 0.00711 0.132 0.992  
## Residuals 92 4.964 0.05396   
## 6 observations deleted due to missingness

#tukeys  
TukeyHSD(vessel\_aov)

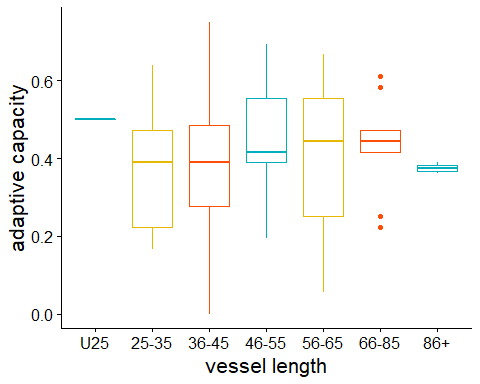
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_vulnerability ~ vessel, data = responses)  
##   
## $vessel  
## diff lwr upr p adj  
## 25-35-U25 0.143113426 -1.2411600 1.5273869 0.9999229  
## 36-45-U25 0.131619385 -1.2390389 1.5022777 0.9999500  
## 46-55-U25 0.011265432 -1.4184061 1.4409369 1.0000000  
## 56-65-U25 0.129647436 -1.2778573 1.5371521 0.9999609  
## 66-85-U25 0.190586420 -1.2390851 1.6202579 0.9996605  
## 86+-U25 0.077777778 -1.5833504 1.7389059 0.9999993  
## 36-45-25-35 -0.011494041 -0.3517707 0.3287826 0.9999999  
## 46-55-25-35 -0.131847994 -0.6619844 0.3982884 0.9890096  
## 56-65-25-35 -0.013465990 -0.4805350 0.4536030 1.0000000  
## 66-85-25-35 0.047472994 -0.4826634 0.5776094 0.9999669  
## 86+-25-35 -0.065335648 -1.0635494 0.9328781 0.9999948  
## 46-55-36-45 -0.120353953 -0.6138473 0.3731394 0.9900904  
## 56-65-36-45 -0.001971949 -0.4269950 0.4230511 1.0000000  
## 66-85-36-45 0.058967034 -0.4345264 0.5524604 0.9998202  
## 86+-36-45 -0.053841608 -1.0330872 0.9254040 0.9999982  
## 56-65-46-55 0.118382004 -0.4697515 0.7065155 0.9964940  
## 66-85-46-55 0.179320988 -0.4600475 0.8186895 0.9795142  
## 86+-46-55 0.066512346 -0.9937604 1.1267851 0.9999960  
## 66-85-56-65 0.060938984 -0.5271945 0.6490725 0.9999219  
## 86+-56-65 -0.051869658 -1.0820576 0.9783183 0.9999989  
## 86+-66-85 -0.112808642 -1.1730814 0.9474641 0.9999087

TukeyHSD(euc\_vessel\_aov)

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_vulnerability\_euc ~ vessel, data = responses)  
##   
## $vessel  
## diff lwr upr p adj  
## 25-35-U25 0.042572634 -0.6734203 0.7585656 0.9999970  
## 36-45-U25 0.034498145 -0.6736695 0.7426658 0.9999991  
## 46-55-U25 -0.028949548 -0.7716820 0.7137829 0.9999998  
## 56-65-U25 0.007400588 -0.7192883 0.7340894 1.0000000  
## 66-85-U25 0.027396880 -0.7107366 0.7655304 0.9999998  
## 86+-U25 -0.019400443 -0.8770340 0.8382332 1.0000000  
## 36-45-25-35 -0.008074489 -0.1909226 0.1747736 0.9999995  
## 46-55-25-35 -0.071522182 -0.3606304 0.2175860 0.9892082  
## 56-65-25-35 -0.035172046 -0.2801388 0.2097947 0.9994733  
## 66-85-25-35 -0.015175754 -0.2922553 0.2619038 0.9999982  
## 86+-25-35 -0.061973077 -0.5791456 0.4551994 0.9998150  
## 46-55-36-45 -0.063447693 -0.3325928 0.2056974 0.9916456  
## 56-65-36-45 -0.027097557 -0.2481501 0.1939550 0.9997889  
## 66-85-36-45 -0.007101265 -0.2632820 0.2490795 1.0000000  
## 86+-36-45 -0.053898588 -0.5601820 0.4523849 0.9999073  
## 56-65-46-55 0.036350136 -0.2783154 0.3510156 0.9998506  
## 66-85-46-55 0.056346428 -0.2839163 0.3966091 0.9988108  
## 86+-46-55 0.009549105 -0.5440510 0.5631492 1.0000000  
## 66-85-56-65 0.019996292 -0.2836546 0.3236472 0.9999946  
## 86+-56-65 -0.026801031 -0.5586828 0.5050808 0.9999989  
## 86+-66-85 -0.046797323 -0.5942118 0.5006171 0.9999745

##anova for ac by vessel length

vessel\_ac<-ggboxplot(responses, x = "vessel", y = "indv\_ac",  
 color = "vessel", palette = c("#00AFBB", "#E7B800", "#FC4E07", "#00AFBB", "#E7B800", "#FC4E07", "#00AFBB"),  
 order = c("U25", "25-35", "36-45", "46-55", "56-65", "66-85", "86+"),  
 ylab = "adaptive capacity", xlab = "vessel length") +  
 theme(plot.title = element\_text(size = 22), axis.title = element\_text(size = 16), legend.position = "none")  
   
vessel\_ac



ggsave(plot = vessel\_ac, file = paste0("../figures/vulnerability/vesselac.png"))

## Saving 5 x 4 in image

#anova  
ac\_vessel\_aov<-aov(indv\_ac ~ vessel, data = responses)  
summary(ac\_vessel\_aov)

## Df Sum Sq Mean Sq F value Pr(>F)  
## vessel 6 0.0915 0.01524 0.579 0.746  
## Residuals 98 2.5815 0.02634

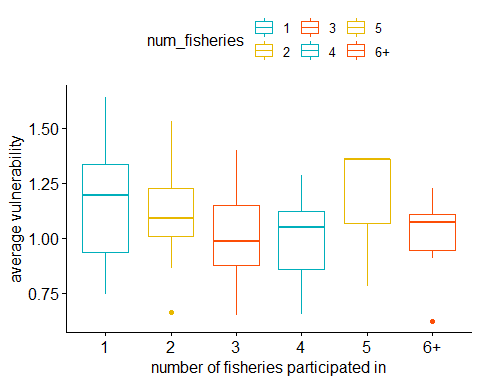
#tukey HSD of anova  
ac\_vessel\_tukey<-TukeyHSD(ac\_vessel\_aov)  
ac\_vessel\_tukey

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_ac ~ vessel, data = responses)  
##   
## $vessel  
## diff lwr upr p adj  
## 25-35-U25 -0.1250000000000014988011 -0.6236766 0.3736766 0.9885464  
## 36-45-U25 -0.1264775413711598517352 -0.6202493 0.3672943 0.9871701  
## 46-55-U25 -0.0493827160493842121092 -0.5644137 0.4656482 0.9999505  
## 56-65-U25 -0.0790598290598306108201 -0.5861053 0.4279857 0.9991647  
## 66-85-U25 -0.0679012345679028128664 -0.5829322 0.4471297 0.9996816  
## 86+-U25 -0.1250000000000016653345 -0.7234119 0.4734119 0.9957015  
## 36-45-25-35 -0.0014775413711583529341 -0.1240603 0.1211052 1.0000000  
## 46-55-25-35 0.0756172839506172866919 -0.1153613 0.2665959 0.8956136  
## 56-65-25-35 0.0459401709401708879810 -0.1223188 0.2141991 0.9821406  
## 66-85-25-35 0.0570987654320986859346 -0.1338798 0.2480774 0.9717482  
## 86+-25-35 -0.0000000000000001665335 -0.3596008 0.3596008 1.0000000  
## 46-55-36-45 0.0770948253217756396261 -0.1006833 0.2548730 0.8478372  
## 56-65-36-45 0.0474177123113292409151 -0.1056944 0.2005298 0.9663882  
## 66-85-36-45 0.0585763068032570388688 -0.1192019 0.2363545 0.9546484  
## 86+-36-45 0.0014775413711581864007 -0.3512901 0.3542452 1.0000000  
## 56-65-46-55 -0.0296771130104463987109 -0.2415488 0.1821946 0.9995483  
## 66-85-46-55 -0.0185185185185186007573 -0.2488474 0.2118103 0.9999825  
## 86+-46-55 -0.0756172839506174532254 -0.4575745 0.3063399 0.9968008  
## 66-85-56-65 0.0111585944919277979537 -0.2007131 0.2230303 0.9999986  
## 86+-56-65 -0.0459401709401710545144 -0.4170595 0.3251791 0.9997791  
## 86+-66-85 -0.0570987654320988524681 -0.4390560 0.3248584 0.9993435

##how many fisheries are people participating in and how does that affect vulnerability and adpative capacity

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 6 x 4  
## num\_fisheries mean count sd  
## <chr> <dbl> <int> <dbl>  
## 1 1 1.17 34 0.258  
## 2 2 1.11 25 0.201  
## 3 3 1.01 16 0.217  
## 4 4 0.993 15 0.192  
## 5 5 1.17 3 0.334  
## 6 6+ 1.02 12 0.157



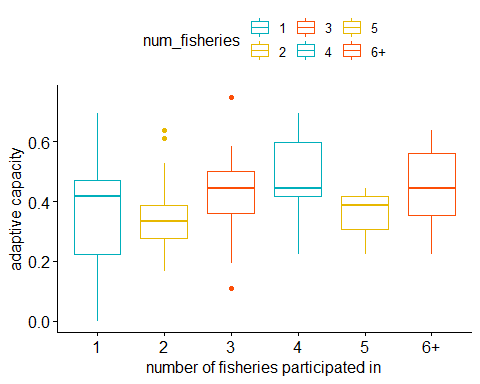
## Saving 5 x 4 in image

## Df Sum Sq Mean Sq F value Pr(>F)   
## num\_fisheries 5 0.502 0.10044 2.074 0.0756 .  
## Residuals 93 4.505 0.04844   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 6 observations deleted due to missingness

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_vulnerability\_euc ~ num\_fisheries, data = responses)  
##   
## $num\_fisheries  
## diff lwr upr p adj  
## 2-1 -0.0525517997 -0.2292839 0.12418034 0.9537624  
## 3-1 -0.1551291263 -0.3545770 0.04431878 0.2197297  
## 4-1 -0.1742993592 -0.3779866 0.02938793 0.1376728  
## 5-1 -0.0001949064 -0.3886113 0.38822151 1.0000000  
## 6+-1 -0.1484379201 -0.3682667 0.07139085 0.3702642  
## 3-2 -0.1025773266 -0.3092802 0.10412559 0.7001161  
## 4-2 -0.1217475595 -0.3325440 0.08904888 0.5478989  
## 5-2 0.0523568933 -0.3398343 0.44454810 0.9988192  
## 6+-2 -0.0958861204 -0.3223178 0.13054558 0.8199118  
## 4-3 -0.0191702328 -0.2493449 0.21100439 0.9998823  
## 5-3 0.1549342200 -0.2480039 0.55787238 0.8723735  
## 6+-3 0.0066912062 -0.2378830 0.25126539 0.9999995  
## 5-4 0.1741044528 -0.2309489 0.57915779 0.8104733  
## 6+-4 0.0258614391 -0.2221821 0.27390494 0.9996450  
## 6+-5 -0.1482430137 -0.5616488 0.26516282 0.9018674

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 6 x 4  
## num\_fisheries mean count sd  
## <chr> <dbl> <int> <dbl>  
## 1 1 0.350 34 0.182  
## 2 2 0.359 25 0.136  
## 3 3 0.429 16 0.150  
## 4 4 0.472 15 0.152  
## 5 5 0.352 3 0.116  
## 6 6+ 0.447 12 0.138



## Saving 5 x 4 in image

## Df Sum Sq Mean Sq F value Pr(>F)   
## num\_fisheries 5 0.2473 0.04947 2.019 0.0825 .  
## Residuals 99 2.4256 0.02450   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

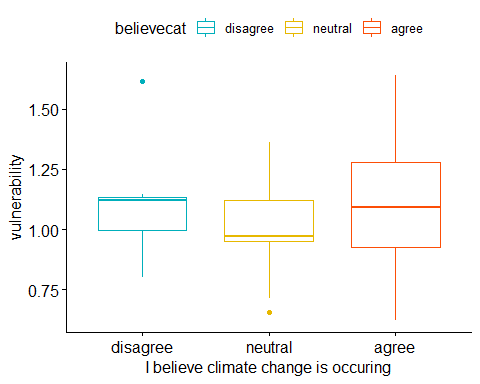
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_ac ~ num\_fisheries, data = responses)  
##   
## $num\_fisheries  
## diff lwr upr p adj  
## 2-1 0.009215686 -0.11063764 0.1290690 0.9999213  
## 3-1 0.079146242 -0.05877117 0.2170637 0.5561255  
## 4-1 0.122549020 -0.01845987 0.2635579 0.1265959  
## 5-1 0.002178649 -0.27181110 0.2761684 1.0000000  
## 6+-1 0.097086057 -0.05566435 0.2498365 0.4407710  
## 3-2 0.069930556 -0.07571441 0.2155755 0.7296662  
## 4-2 0.113333333 -0.03524241 0.2619091 0.2394270  
## 5-2 -0.007037037 -0.28499680 0.2709227 0.9999997  
## 6+-2 0.087870370 -0.07189174 0.2476325 0.6015074  
## 4-3 0.043402778 -0.12009384 0.2068994 0.9716795  
## 5-3 -0.076967593 -0.36318084 0.2092457 0.9700498  
## 6+-3 0.017939815 -0.15578504 0.1916647 0.9996641  
## 5-4 -0.120370370 -0.40808606 0.1673453 0.8281414  
## 6+-4 -0.025462963 -0.20165212 0.1507262 0.9982844  
## 6+-5 0.094907407 -0.19874119 0.3885560 0.9352634

##varibility in vulnerability depending on if you think climate change is happening or not

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 3 x 6  
## believecat meanex meansen meanac meanvuln count  
## <chr> <dbl> <dbl> <dbl> <dbl> <int>  
## 1 agree 0.676 0.555 0.399 1.10 81  
## 2 disagree 0.657 0.619 0.389 1.11 7  
## 3 neutral 0.498 0.567 0.363 1.01 17

## Warning: Removed 6 rows containing non-finite values (stat\_boxplot).



## Saving 5 x 4 in image

## Warning: Removed 6 rows containing non-finite values (stat\_boxplot).

## Df Sum Sq Mean Sq F value Pr(>F)  
## believecat 2 0.119 0.05926 1.164 0.317  
## Residuals 96 4.888 0.05092   
## 6 observations deleted due to missingness

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_vulnerability\_euc ~ believecat, data = responses)  
##   
## $believecat  
## diff lwr upr p adj  
## disagree-agree 0.01383016 -0.1984751 0.22613540 0.9868308  
## neutral-agree -0.09008620 -0.2343882 0.05421577 0.3020403  
## neutral-disagree -0.10391636 -0.3451654 0.13733267 0.5627125

## Df Sum Sq Mean Sq F value Pr(>F)   
## believecat 2 0.440 0.22024 4.955 0.00895 \*\*  
## Residuals 96 4.267 0.04445   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## 6 observations deleted due to missingness

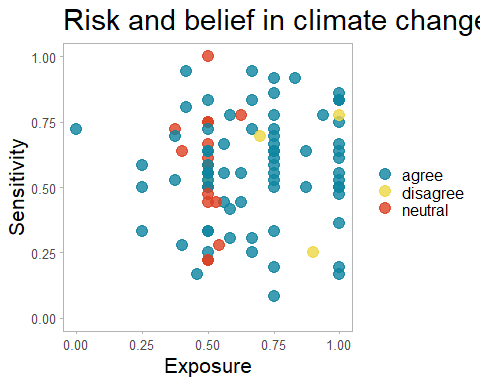
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_exposure ~ believecat, data = responses)  
##   
## $believecat  
## diff lwr upr p adj  
## disagree-agree -0.01930159 -0.2176542 0.17905103 0.9708589  
## neutral-agree -0.17803758 -0.3128561 -0.04321908 0.0062296  
## neutral-disagree -0.15873599 -0.3841302 0.06665824 0.2194312

## Df Sum Sq Mean Sq F value Pr(>F)  
## believecat 2 0.027 0.01345 0.297 0.744  
## Residuals 102 4.621 0.04530

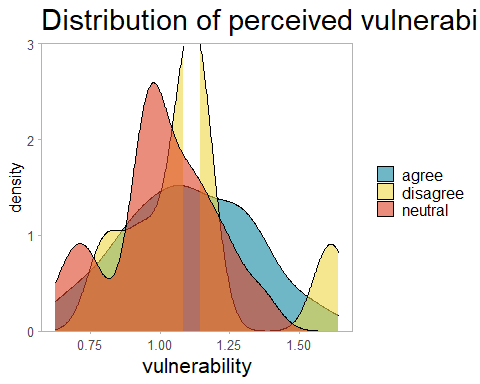
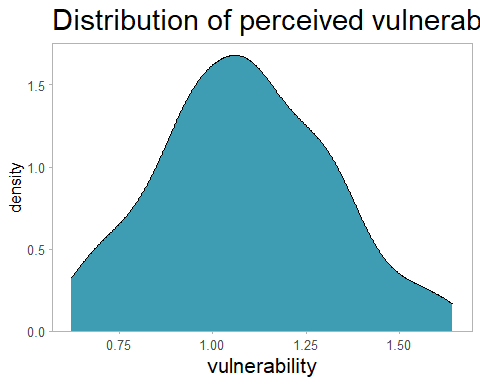
## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_sensitivity ~ believecat, data = responses)  
##   
## $believecat  
## diff lwr upr p adj  
## disagree-agree 0.06383500 -0.1355966 0.2632666 0.7274985  
## neutral-agree 0.01178084 -0.1232678 0.1468295 0.9765538  
## neutral-disagree -0.05205415 -0.2793944 0.1752861 0.8494687

## Df Sum Sq Mean Sq F value Pr(>F)  
## believecat 2 0.0188 0.009376 0.36 0.698  
## Residuals 102 2.6542 0.026021

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_ac ~ believecat, data = responses)  
##   
## $believecat  
## diff lwr upr p adj  
## disagree-agree -0.01028807 -0.1614360 0.14085987 0.9856574  
## neutral-agree -0.03643186 -0.1387844 0.06592065 0.6750150  
## neutral-disagree -0.02614379 -0.1984435 0.14615593 0.9307785

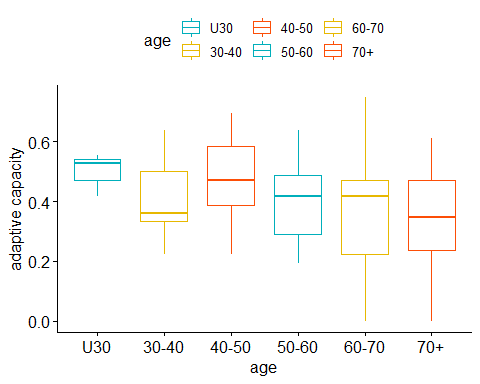
##scatterplot of risk depending on your belief in climate change 

## Saving 5 x 4 in image

##distribution of vulnerability ##vulnerability plots used in tnc talk 

##anova for adaptive capacity by age

## Saving 5 x 4 in image



## Df Sum Sq Mean Sq F value Pr(>F)  
## age 5 0.1991 0.03982 1.594 0.169  
## Residuals 99 2.4738 0.02499

## Tukey multiple comparisons of means  
## 95% family-wise confidence level  
##   
## Fit: aov(formula = indv\_ac ~ age, data = responses)  
##   
## $age  
## diff lwr upr p adj  
## 40-50-30-40 0.07799145 -0.1152500 0.27123291 0.8486153  
## 50-60-30-40 -0.00462963 -0.1746994 0.16544011 0.9999996  
## 60-70-30-40 -0.03518519 -0.2029409 0.13257057 0.9901178  
## 70+-30-40 -0.05429293 -0.2295081 0.12092224 0.9454937  
## U30-30-40 0.09722222 -0.2052038 0.39964821 0.9366525  
## 50-60-40-50 -0.08262108 -0.2377113 0.07246917 0.6341360  
## 60-70-40-50 -0.11317664 -0.2657259 0.03937258 0.2678021  
## 70+-40-50 -0.13228438 -0.2930004 0.02843160 0.1690679  
## U30-40-50 0.01923077 -0.2750323 0.31349383 0.9999648  
## 60-70-50-60 -0.03055556 -0.1524273 0.09131617 0.9779444  
## 70+-50-60 -0.04966330 -0.1816144 0.08228776 0.8827503  
## U30-50-60 0.10185185 -0.1777411 0.38144478 0.8963543  
## 70+-60-70 -0.01910774 -0.1480626 0.10984713 0.9980664  
## U30-60-70 0.13240741 -0.1457840 0.41059885 0.7368987  
## U30-70+ 0.15151515 -0.1312371 0.43426742 0.6282539

#save with vulnerability calculations