

Analysis of human data

Marieke Jones, PhD

2022-11-02

Contents

Set up	2
Analysis Goals:	2
Load and clean the data sets for each brain	3
Question 1 - Length	5
FIGURE: Length by status	7
FIGURE: AIS Length by Status and Tangle	9
Question 2 - Distribution of TRIM46 across distance	11
FIGURE: Distance by Intensity for each brain	11
FIGURE: Splines model plot	17
Question 3 - Mean Intensity	18
FIGURE: Mean Intensity by status	21
FIGURE: Mean Intensity by Status and Tangle	22
Question 4 & 5 - Max and Min Intensity	24
Max intensity	25
FIGURE: Max intensity by status	28
FIGURE: Max intensity by status and tangle	29
Min intensity	31
FIGURE: Min intensity by status	33
FIGURE: Min intensity by status and tangle	35
Question 6 - AD only Length by severity of disease	37

Question 7 - AD only AIS intensity by severity	40
Question 7a - Splines for intensity by severity	41
Question 7b - Mean intensity by severity	42
Question 7c - Max intensity by severity	44

Set up

Load libraries

```
library(tidyverse)
library(purrr)
library(stringr)
library(readxl)
library(lme4)
library(lmerTest)
library(modelr)
library(zoo)
library(ggeffects)
library(emmeans)

library(splines)
```

Analysis Goals:

1. Is there a difference in length of the axon initial segment between AD and non-AD neurons? Is the length affected by the presence of tangles?
2. What does the TRIM46 fluorescence look like across the axon initial segment for AD v. non-AD neurons? Is the shape influenced by presence of tangles?
3. Does mean AIS intensity at each distance differ between AD and non-AD neurons? Is the mean AIS intensity impacted by the presence of a tangle?
4. Is the maximum AIS intensity for each neuron different between AD and non-AD brains? Is the maximum AIS intensity affected by the presence of tangles?
5. Is the minimum AIS intensity for each neuron different between AD and non-AD brains? Is the minimum AIS intensity affected by the presence of tangles?
6. Within Alzheimer's brains, is there a difference in AIS length across severity of disease?
7. Within Alzheimer's brains, is there a difference in AIS intensity across severity of disease?
 - a. Splines
 - b. Mean
 - c. Max

Load and clean the data sets for each brain

Each brain's data is stored in a separate file where the first column is the distance from the start of the axon initial segment and the other columns are the fluorescence intensity of TRIM46 (protein present in axon initial segment that is important for brain function) in a given neuron.

Write a function that will read the file and clean it

```
readclean <- function(filename){
  dat <- read_excel(paste0("Data/Human/5D - Mean Concentration _ 5E - AIS Length/", filename), skip = 1)
  dplyr::select(-avg, -std, -count, - `std rror`) %>%
  mutate(filename = filename) %>%
  pivot_longer(cols = contains("Neur"), names_to = "Neuron", values_to = "TRIM46_intensity")
}
```

Create a list of all of the datafiles and then iterate through them to read them all in.

```
my_files <- list.files(path = "Data/Human/5D - Mean Concentration _ 5E - AIS Length/")

dfs <- map(my_files, ~readclean(.))
alldat <- bind_rows(dfs)

alldat
```

```
## # A tibble: 101,945 x 4
##   `Distance_(microns)` filename      Neuron      TRIM46_intensity
##   <dbl> <chr>                <chr>      <dbl>
## 1      0 5D 5E - AD 1 = A16-109.xlsx Neuron 1      144.
## 2      0 5D 5E - AD 1 = A16-109.xlsx Neuron 2      160
## 3      0 5D 5E - AD 1 = A16-109.xlsx Neuron 3      144
## 4      0 5D 5E - AD 1 = A16-109.xlsx Neuron 4      161
## 5      0 5D 5E - AD 1 = A16-109.xlsx Neuron 5      139.
## 6      0 5D 5E - AD 1 = A16-109.xlsx Neuron 6      133
## 7      0 5D 5E - AD 1 = A16-109.xlsx Neuron 7      138
## 8      0 5D 5E - AD 1 = A16-109.xlsx Neuron 8      144
## 9      0 5D 5E - AD 1 = A16-109.xlsx Neuron 9      172
## 10     0 5D 5E - AD 1 = A16-109.xlsx Neuron 10     142.
## # ... with 101,935 more rows
```

Clean resulting dataset

Create two new columns to denote Status and Brain

```
alldat <- alldat %>%
  separate(filename, into = c(NA, "filename"), sep = " - ") %>%
  separate(filename, into = c("Status", "Brain"), sep = " = ") %>%
  separate(Status, into = c("Status", NA), sep = " ") %>%
  separate(Brain, into = c("Brain", NA), sep = "\\.")

alldat
```

```
## # A tibble: 101,945 x 5
```

```
##      `Distance_(microns)` Status Brain   Neuron   TRIM46_intensity
##      <dbl> <chr> <chr> <chr> <dbl>
##  1          0 AD    A16-109 Neuron 1      144.
##  2          0 AD    A16-109 Neuron 2      160
##  3          0 AD    A16-109 Neuron 3      144
##  4          0 AD    A16-109 Neuron 4      161
##  5          0 AD    A16-109 Neuron 5     139.
##  6          0 AD    A16-109 Neuron 6      133
##  7          0 AD    A16-109 Neuron 7      138
##  8          0 AD    A16-109 Neuron 8      144
##  9          0 AD    A16-109 Neuron 9      172
## 10          0 AD    A16-109 Neuron 10     142.
## # ... with 101,935 more rows
```

Rename Distance column

```
alldat <- alldat %>%
  rename(Dist = `Distance_(microns)`)
```

Remove rows where TRIM46 Intensity is missing since those neurons were not that long.

```
alldat <- alldat %>%
  arrange(Brain, Neuron, Dist) %>%
  drop_na(TRIM46_intensity)
```

Create new column that designates whether or not Neuron has a tangle or tau accumulation

```
alldat %>% count(Brain, Neuron) #1160 unique neurons
```

```
## # A tibble: 1,160 x 3
##   Brain   Neuron      n
##   <chr> <chr>    <int>
## 1 A11-30 Neuron 1 (tang)    14
## 2 A11-30 Neuron 10        11
## 3 A11-30 Neuron 100       25
## 4 A11-30 Neuron 101       30
## 5 A11-30 Neuron 102 (ta)   19
## 6 A11-30 Neuron 103 (tang) 12
## 7 A11-30 Neuron 104 (ta)   21
## 8 A11-30 Neuron 105 (ta)   19
## 9 A11-30 Neuron 106       12
## 10 A11-30 Neuron 107 (tang) 13
## # ... with 1,150 more rows
```

```
alldat %>% count(Brain, Neuron) %>% arrange(-n)
```

```
## # A tibble: 1,160 x 3
##   Brain   Neuron      n
##   <chr> <chr>    <int>
## 1 A14-3   Neuron 101    89
## 2 A16-109 Neuron 33     72
```

```
## 3 A15-167_ Neuron 6      71
## 4 A14-3    Neuron 95     70
## 5 A16-109  Neuron 43     70
## 6 A14-3    Neuron 92     69
## 7 A13-66   Neuron 158 (ta) 65
## 8 A14-3    Neuron 104     64
## 9 A14-3    Neuron 25      64
## 10 A14-3   Neuron 70      64
## # ... with 1,150 more rows
```

Create new column that designates whether or not Neuron has a tangle or ta (tau accumulation)

```
alldat <- alldat %>%
  mutate(Tangle = case_when(str_detect(Neuron, "(tang)") ~ "tangle",
                             str_detect(Neuron, "(ta)") ~ "ta",
                             TRUE ~ "none"))

alldat %>% count(Status, Tangle)
```

```
## # A tibble: 6 x 3
##   Status Tangle     n
##   <chr>  <chr> <int>
## 1 AD     none   11233
## 2 AD     ta     2207
## 3 AD     tangle  1478
## 4 Non-AD none   9248
## 5 Non-AD ta     3496
## 6 Non-AD tangle  118
```

There are very few neurons from non-AD brains with a tangle.

Set Non-AD Status to be first

```
alldat <- alldat %>%
  mutate(Status = factor(Status, levels = c("Non-AD", "AD")))

alldat %>% count(Status)
```

```
## # A tibble: 2 x 2
##   Status     n
##   <fct> <int>
## 1 Non-AD 12862
## 2 AD    14918
```

Question 1 - Length

Is there a difference in length of the axon initial segment between AD and non-AD neurons? Is the length affected by the presence of tangles?

Note: Because of technical limitations, we don't know if we have examined the entire axon initial segment. However, the technical limitation was the same for AD and non-AD tissue, so it should be equally a problem across all tissue.

Create dataset that is the maximum distance of each neuron aka the length

```
maxdist <- alldat %>%
  group_by(Brain, Neuron) %>%
  filter(Dist == max(Dist)) %>%
  ungroup()
```

Each neuron has one maximum distance (aka length) but each person has several maxima. Therefore, use linear mixed model to compare.

Merci is excited to investigate the effect of Tangles on length of Axon Initial Segment in Alzheimer's so add in an interaction with Tangle.

```
maxdistmod <- lmer(Dist ~ Status*Tangle + (1|Brain), data = maxdist)
summary(maxdistmod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Dist ~ Status * Tangle + (1 | Brain)
## Data: maxdist
##
## REML criterion at convergence: 4852.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.0366 -0.7463 -0.2028  0.4972  4.8512
##
## Random effects:
## Groups Name Variance Std.Dev.
## Brain (Intercept) 0.2005  0.4478
## Residual 3.7638  1.9401
## Number of obs: 1160, groups: Brain, 16
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    4.0649    0.2003   17.4416  20.292 1.42e-13 ***
## StatusAD      -0.3541    0.2726   15.4747  -1.299  0.2129
## Tangleta      -0.3927    0.2408  328.3683  -1.631  0.1039
## TangleTangle  -1.4842    0.7112 1148.7532  -2.087  0.0371 *
## StatusAD:Tangleta  0.1341    0.3367  585.2851   0.398  0.6905
## StatusAD:TangleTangle 0.4586    0.7520 1143.9364   0.610  0.5421
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) SttsAD Tngltn StatsAD:Tngltn
## StatusAD      -0.735
## Tangleta      -0.389  0.286
## TangleTangle  -0.121  0.089  0.180
## StatsAD:Tngltn 0.278 -0.300 -0.715 -0.129
## SttsAD:Tngltn  0.114 -0.129 -0.171 -0.946  0.188
```

This model output shows raw t-tests. Better to perform testing with Tukey correction for the number of tests performed.

```
pairs(emmeans(maxdistmod, specs = "Tangle", by = "Status"))
```

```
## Status = Non-AD:
## contrast      estimate    SE    df t.ratio p.value
## none - ta      0.393 0.246   342   1.596  0.2489
## none - tangle   1.484 0.713  1149   2.082  0.0941
## ta - tangle     1.091 0.709  1149   1.539  0.2728
##
## Status = AD:
## contrast      estimate    SE    df t.ratio p.value
## none - ta      0.259 0.237  1030   1.091  0.5202
## none - tangle   1.026 0.246  1055   4.166  0.0001
## ta - tangle     0.767 0.287  1150   2.676  0.0206
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

According to Tukey, AD neurons with tangles are 1.026 microns shorter than AD neurons with no tangle ($p = 0.0001$)

Get pairwise comparisons within Tangle levels

```
pairs(emmeans(maxdistmod, specs = "Status", by = "Tangle"))
```

```
## Tangle = none:
## contrast      estimate    SE    df t.ratio p.value
## (Non-AD) - AD   0.354 0.273   16.4   1.295  0.2133
##
## Tangle = ta:
## contrast      estimate    SE    df t.ratio p.value
## (Non-AD) - AD   0.220 0.367   43.3   0.599  0.5520
##
## Tangle = tangle:
## contrast      estimate    SE    df t.ratio p.value
## (Non-AD) - AD  -0.104 0.768  541.9  -0.136  0.8919
##
## Degrees-of-freedom method: kenward-roger
```

According to Tukey pairwise comparisons, within neurons with a tangle, there is no difference in length between non-AD and AD ($p = 0.892$). AD with tangle are 0.104 microns shorter than non-AD

FIGURE: Length by status

Each dot is the length for one neuron.

On top of dots are the predicted means and 95% confidence interval from the linear mixed model.

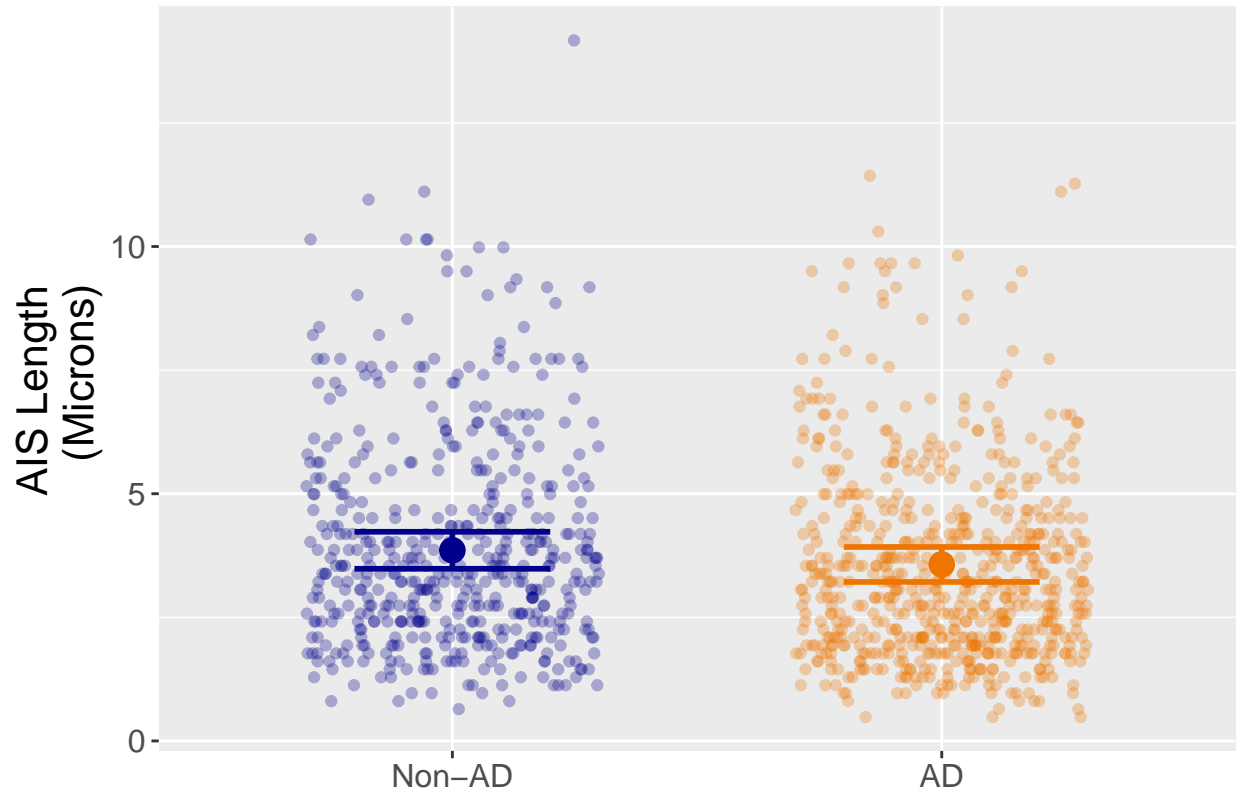
```
# return dataset of predicted mean +/- 95%CI
pred_mean_lengthmod_status <- ggeffect(maxdistmod, terms = "Status") %>%
  as_tibble() %>%
  rename(Status = x)

pred_mean_lengthmod_status
```

```
## # A tibble: 2 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>    <dbl>   <dbl>   <dbl> <fct>
## 1 Non-AD      3.86     0.191    3.48    4.23  1
## 2 AD          3.57     0.180    3.22    3.92  1
```

```
lengthplotstatus <- maxdist %>%
  ggplot() +
  geom_jitter(aes(x = Status,
                  y = Dist,
                  color = Status),
              alpha = .3,
              width = .3) +
  geom_point(data = pred_mean_lengthmod_status,
             aes(x = Status,
                 y = predicted,
                 color = Status),
             size = 4) +
  geom_errorbar(data = pred_mean_lengthmod_status,
               aes(x = Status,
                   ymin = conf.low,
                   ymax = conf.high,
                   color = Status),
               width = .4,
               lwd = 1) +
  labs(y = "AIS Length \n (Microns)", x = "") +
  scale_color_manual(values = c("darkblue", "darkorange2")) +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12))

lengthplotstatus
```

```
ggsave(lengthplotstatus, filename = "Figures/human_length_status.png", width = 6, height = 4)
```

FIGURE: AIS Length by Status and Tangle

Show jitter plus predicted mean and 95% CI with these colors.

```
# return dataset of predicted mean +/- 95%CI
pred_mean_lengthmod <- ggeffect(maxdistmod, terms = c("Status", "Tangle")) %>%
  as_tibble() %>%
  rename(Status = x)
```

```
pred_mean_lengthmod
```

```
## # A tibble: 6 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>    <dbl>   <dbl>   <dbl> <fct>
## 1 Non-AD     4.06    0.200    3.67    4.46 none
## 2 AD         3.71    0.185    3.35    4.07 none
## 3 Non-AD     3.67    0.246    3.19    4.16 ta
## 4 AD         3.45    0.268    2.93    3.98 ta
## 5 Non-AD     2.58    0.715    1.18    3.98 tangle
## 6 AD         2.69    0.275    2.15    3.22 tangle
```

```

lengthplot <- maxdist %>%
  ggplot() +
  geom_point(aes(x = Status,
                 y = Dist,
                 color = Tangle),
             position = position_jitterdodge(jitter.width = .2),
             alpha = .2) +
  geom_point(data = pred_mean_lengthmod,
             aes(x = Status,
                 y = predicted,
                 color = group),
             position = position_dodge(width = .75),
             size = 4) +
  geom_errorbar(data = pred_mean_lengthmod,
               aes(x = Status,
                   ymin = conf.low,
                   ymax = conf.high,
                   color = group),
               position = position_dodge(width = .75),
               width = .6,
               lwd = 1) +
  labs(y = "AIS Length \n (Microns)", x = "") +
  scale_color_manual(values = c("darkorchid", "darkcyan", "gold")) +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12))

lengthplot

```



```
ggsave(lengthplot, filename = "Figures/human_length_status_tangle.png", width = 6, height = 4)
```

Question 2 - Distribution of TRIM46 across distance

What does the TRIM46 fluorescence look like across the axon initial segment for AD v. non-AD neurons? Is the shape influenced by presence of tangles?

To show the shape of the TRIM46 intensity across distance, the raw data looked very messy. A rolling average across 3-distance observations works nicely to show the trend.

I used the `rollapply()` function rather than the more standard `rollmean()` function because `rollmean()` has no way to remove NAs.

```
alldat <- alldat %>%
  group_by(Brain, Neuron) %>%
  mutate(roll_Intensity = rollapply(TRIM46_intensity, 3, mean, na.rm = TRUE, fill = NA)) %>%
  ungroup()
```

FIGURE: Distance by Intensity for each brain

One line per brain, rolling average averaged over all neurons at a given dist.

The rolling average is the average of a 3-dist chunk.

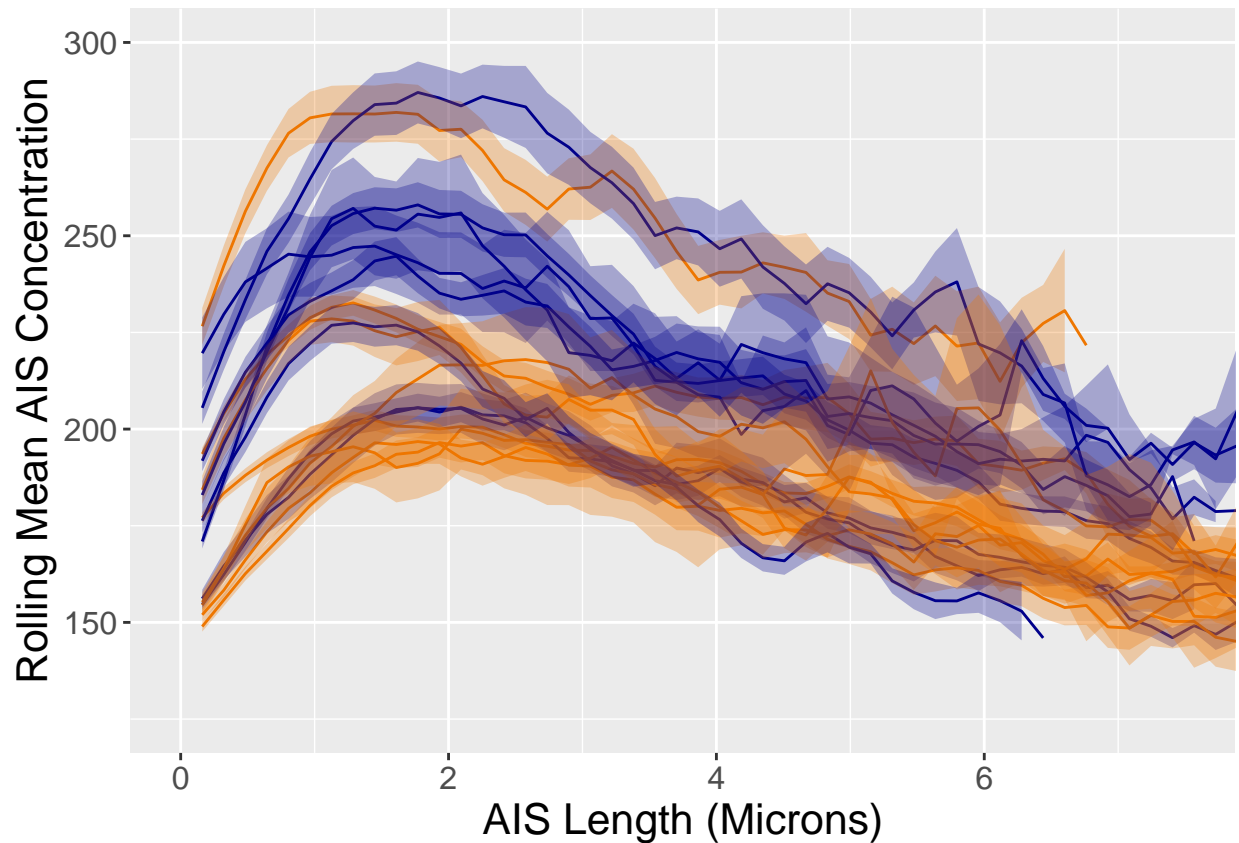
```
rollavgdat <- alldat %>%
  group_by(Status, Brain, Dist) %>%
  summarize(Mean_Intensity = mean(roll_Intensity, na.rm = TRUE),
            sd_Intensity = sd(roll_Intensity, na.rm = TRUE),
            n_neurons = n(),
            se_Intensity = sd_Intensity/sqrt(n_neurons))
```

`summarise()` has grouped output by 'Status', 'Brain'. You can override using
the `.groups` argument.

```
lineplot <- rollavgdat %>%
  ggplot(aes(Dist, Mean_Intensity)) +
  geom_line(aes(group = Brain, color = Status)) +
  geom_ribbon(aes(ymin = Mean_Intensity-se_Intensity,
                ymax = Mean_Intensity+se_Intensity,
                group = Brain, fill = Status), alpha = .3) +
  scale_fill_manual(values = c("darkblue", "darkorange2")) +
  scale_color_manual(values = c("darkblue", "darkorange2")) +
  labs(y = "Rolling Mean AIS Concentration", x = "AIS Length (Microns)") +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12)) +
  coord_cartesian(xlim = c(0, 7.5),
                  ylim = c(125, 300))
```

lineplot

Warning: Removed 32 row(s) containing missing values (geom_path).



```
ggsave(lineplot, filename = "Figures/human_lineplot_distVintensity.png", width = 6, height = 4)
```

```
## Warning: Removed 32 row(s) containing missing values (geom_path).
```

Splines are a way to fit a non-linear curve to data to understand how the relationship between Distance and Intensity changes for Alz v. Normal and based on tangle status.

Try basic natural splines model with no interactions to just understand the splines portion of the model

```
splinemod <- lmer(TRIM46_intensity ~ ns(Dist, df = 5) + Status + Tangle + (1|Brain), data = alldat)
```

Now try with interaction terms

```
splinemod2 <- lmer(TRIM46_intensity ~ ns(Dist, df = 5)*Status + Status*Tangle + ns(Dist, df = 5)*Tangle
```

Compare splinemod1 and splinemod2

```
anova(splinemod, splinemod2)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: alldat
```

```
## Models:
```

```
## splinemod: TRIM46_intensity ~ ns(Dist, df = 5) + Status + Tangle + (1 | Brain)
## splinemod2: TRIM46_intensity ~ ns(Dist, df = 5) * Status + Status * Tangle + ns(Dist, df = 5) * Tang
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## splinemod      11 287979 288069 -143978   287957
## splinemod2     28 287546 287776 -143745   287490 467.18 17 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

splinemod2 is better.

Try to simplify the model by removing the interaction of tangle and dist

```
splinemod3 <- lmer(TRIM46_intensity ~ ns(Dist, df = 5)*Status + Status*Tangle + (1|Brain), data = allda
```

See if there is a difference in model fit between splinemod3 and splinemod2

```
anova(splinemod2, splinemod3)
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: alldat
## Models:
## splinemod3: TRIM46_intensity ~ ns(Dist, df = 5) * Status + Status * Tangle + (1 | Brain)
## splinemod2: TRIM46_intensity ~ ns(Dist, df = 5) * Status + Status * Tangle + ns(Dist, df = 5) * Tang
##           npar      AIC      BIC logLik deviance Chisq Df Pr(>Chisq)
## splinemod3     18 287626 287775 -143795   287590
## splinemod2     28 287546 287776 -143745   287490 100.86 10 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

splinemod2 fits better than splinemod3, so use that

```
summary(splinemod2)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: TRIM46_intensity ~ ns(Dist, df = 5) * Status + Status * Tangle +
##          ns(Dist, df = 5) * Tangle + (1 | Brain)
## Data: alldat
##
## REML criterion at convergence: 287371
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.3197 -0.6383 -0.1099  0.4730 12.5693
##
## Random effects:
##  Groups   Name                Variance Std.Dev.
## Brain    (Intercept)         579.6    24.08
## Residual                    1823.1    42.70
## Number of obs: 27780, groups: Brain, 16
##
```

```

## Fixed effects:
##
##              Estimate Std. Error      df t value
## (Intercept)    162.8714      8.6805    15.0638  18.763
## ns(Dist, df = 5)1      83.6917      2.1377  27740.0275  39.151
## ns(Dist, df = 5)2      63.0806      2.2300  27740.0710  28.287
## ns(Dist, df = 5)3     -28.5533      3.3613  27740.7223  -8.495
## ns(Dist, df = 5)4      33.9968      5.3620  27740.9970   6.340
## ns(Dist, df = 5)5     -51.9338      8.0319  27741.0856  -6.466
## StatusAD         3.7068     12.2185    14.7831   0.303
## Tangleta        11.6806      2.6183  27753.9764   4.461
## Tangletangle      0.9811      5.3381  27745.6134   0.184
## ns(Dist, df = 5)1:StatusAD -29.8862      2.6512  27740.0288 -11.273
## ns(Dist, df = 5)2:StatusAD -18.8716      2.8224  27740.0808  -6.686
## ns(Dist, df = 5)3:StatusAD  14.2739      4.8023  27740.6589   2.972
## ns(Dist, df = 5)4:StatusAD  -7.6481      7.5844  27740.7656  -1.008
## ns(Dist, df = 5)5:StatusAD   7.8479     12.9671  27740.8106   0.605
## StatusAD:Tangleta    -12.5302      1.6805  27706.3513  -7.456
## StatusAD:Tangletangle  1.8960      4.2952  27749.4450   0.441
## ns(Dist, df = 5)1:Tangleta  -3.1770      3.1698  27740.0134  -1.002
## ns(Dist, df = 5)2:Tangleta -11.2710      3.4682  27740.0837  -3.250
## ns(Dist, df = 5)3:Tangleta  12.2471      6.9332  27740.2912   1.766
## ns(Dist, df = 5)4:Tangleta -14.2708     12.2833  27740.4091  -1.162
## ns(Dist, df = 5)5:Tangleta -28.4808     23.8613  27740.3754  -1.194
## ns(Dist, df = 5)1:Tangletangle -27.4960      5.0883  27740.0209  -5.404
## ns(Dist, df = 5)2:Tangletangle -31.8846      6.2452  27740.2040  -5.105
## ns(Dist, df = 5)3:Tangletangle  29.7751     18.3715  27740.3753   1.621
## ns(Dist, df = 5)4:Tangletangle -34.3811     31.6867  27740.0753  -1.085
## ns(Dist, df = 5)5:Tangletangle -87.8557     68.0533  27740.0888  -1.291
##
##              Pr(>|t|)
## (Intercept)    7.38e-12 ***
## ns(Dist, df = 5)1      < 2e-16 ***
## ns(Dist, df = 5)2      < 2e-16 ***
## ns(Dist, df = 5)3      < 2e-16 ***
## ns(Dist, df = 5)4     2.33e-10 ***
## ns(Dist, df = 5)5     1.02e-10 ***
## StatusAD         0.76583
## Tangleta        8.18e-06 ***
## Tangletangle      0.85418
## ns(Dist, df = 5)1:StatusAD < 2e-16 ***
## ns(Dist, df = 5)2:StatusAD 2.33e-11 ***
## ns(Dist, df = 5)3:StatusAD 0.00296 **
## ns(Dist, df = 5)4:StatusAD 0.31327
## ns(Dist, df = 5)5:StatusAD 0.54504
## StatusAD:Tangleta  9.16e-14 ***
## StatusAD:Tangletangle 0.65890
## ns(Dist, df = 5)1:Tangleta 0.31622
## ns(Dist, df = 5)2:Tangleta 0.00116 **
## ns(Dist, df = 5)3:Tangleta 0.07733 .
## ns(Dist, df = 5)4:Tangleta 0.24532
## ns(Dist, df = 5)5:Tangleta 0.23265
## ns(Dist, df = 5)1:Tangletangle 6.58e-08 ***
## ns(Dist, df = 5)2:Tangletangle 3.32e-07 ***
## ns(Dist, df = 5)3:Tangletangle 0.10509
## ns(Dist, df = 5)4:Tangletangle 0.27792

```

```
## ns(Dist, df = 5)5:Tangletangle 0.19672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 26 > 12.
## Use print(x, correlation=TRUE) or
##      vcov(x)          if you need it
```

Interpretation: - At Dist=0, non-AD neurons have average Intensity of 162 - Across Dist, Intensity is not linear - At Dist=0, AD neurons have Intensity 3.2 higher on average ($p = ns$) - Compared to neurons with no tangle in non-AD brains, neurons with ta in non-AD brains have intensity 12.5 lower ($p = 2.13e-13$) - Compared to neurons with no tangle in non-AD brains, neurons with tangle in non-AD brains have intensity 1.8 lower ($p = ns$) - There is a different shape for AD-ta and AD-tangle

```
pairs(emmeans(splinemod2, specs = "Status"))
```

```
## contrast      estimate    SE df z.ratio p.value
## (Non-AD) - AD      21.2 12.2 Inf   1.740  0.0818
##
## Results are averaged over the levels of: Tangle
## Degrees-of-freedom method: asymptotic
```

For all neurons, on average, AD neurons have intensity 21.2 lower than non-AD ($p = 0.082$)

See the difference between AD and non-AD for each level of tangle:

```
pairs(emmeans(splinemod2, specs = "Status", by = "Tangle"))
```

```
## Tangle = none:
## contrast      estimate    SE df z.ratio p.value
## (Non-AD) - AD      17.6 12.1 Inf   1.458  0.1449
##
## Tangle = ta:
## contrast      estimate    SE df z.ratio p.value
## (Non-AD) - AD      30.2 12.1 Inf   2.482  0.0131
##
## Tangle = tangle:
## contrast      estimate    SE df z.ratio p.value
## (Non-AD) - AD      15.7 12.8 Inf   1.231  0.2184
##
## Degrees-of-freedom method: asymptotic
```

- Within neurons with no tangle, AD neurons have intensity 17.6 lower than non-AD ($p = 0.145$)
- Within neurons with ta, AD neurons have intensity 30.2 lower than non-AD ($p = 0.013$)
- Within neurons with tangle, AD neurons have intensity 15.7 lower than non-AD ($p = 0.218$)

See the difference between tangle levels for each Status:

```
pairs(emmeans(splinemod2, specs = "Tangle", by = "Status"))
```



```
## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'pbkrtest.limit = 27780' (or larger)
## [or, globally, 'set emm_options(pbkrtest.limit = 27780)' or larger];
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.
## To enable adjustments, add the argument 'lmerTest.limit = 27780' (or larger)
## [or, globally, 'set emm_options(lmerTest.limit = 27780)' or larger];
## but be warned that this may result in large computation time and memory use.

## Status = Non-AD:
## contrast      estimate    SE df z.ratio p.value
## none - ta      -3.36 1.62 Inf  -2.072  0.0956
## none - tangle   27.97 4.53 Inf   6.180 <.0001
## ta - tangle     31.33 4.56 Inf   6.869 <.0001
##
## Status = AD:
## contrast      estimate    SE df z.ratio p.value
## none - ta       9.17 1.53 Inf   5.984 <.0001
## none - tangle   26.07 2.48 Inf  10.513 <.0001
## ta - tangle     16.90 2.70 Inf   6.257 <.0001
##
## Degrees-of-freedom method: asymptotic
## P value adjustment: tukey method for comparing a family of 3 estimates
```

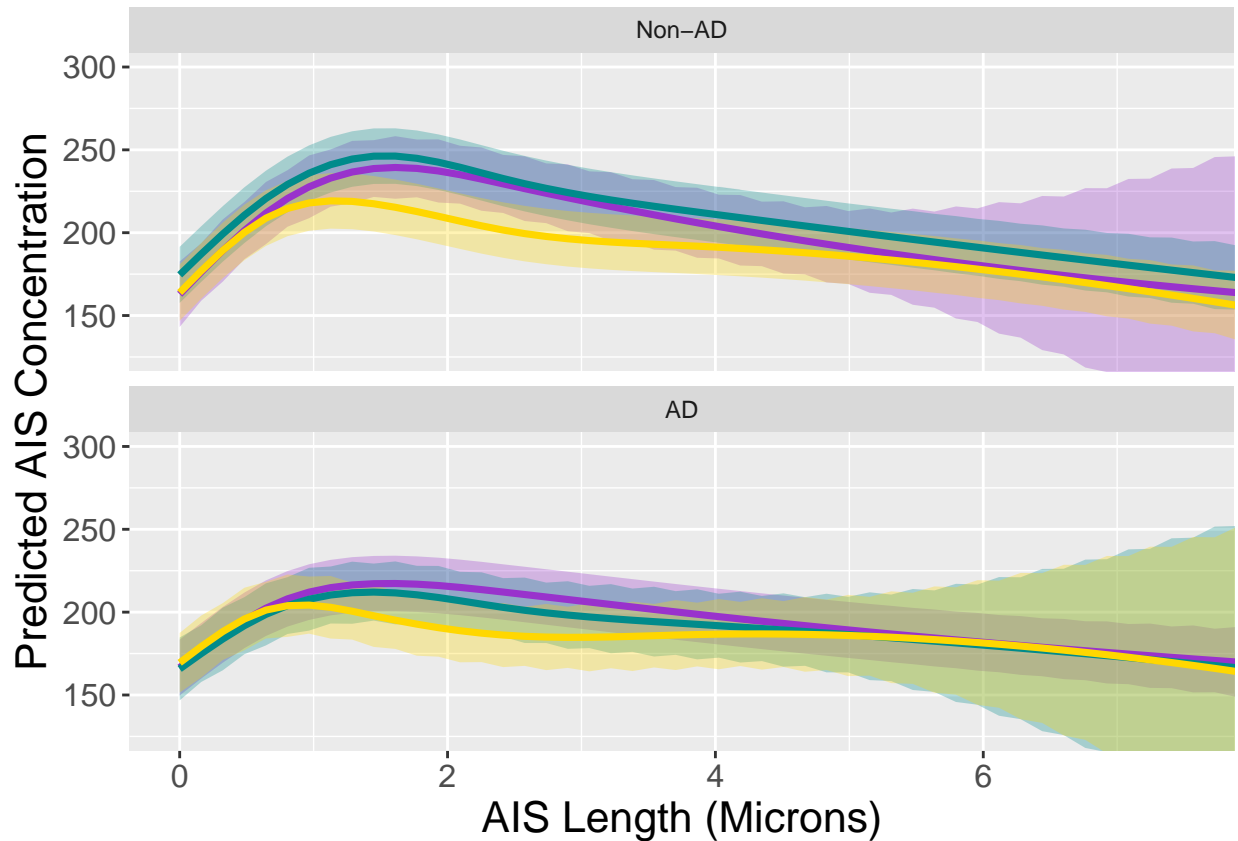
- Within non-AD neurons:
 - ta have intensity 3.3 higher than none ($p = 0.096$)
 - tangles have intensity 27.97 lower than none ($p < 0.0001$)
 - tangles have intensity 31.33 lower than ta ($p < 0.0001$)
- Within AD neurons:
 - ta have intensity 9.17 lower than none ($p < 0.0001$)
 - tangles have intensity 26.07 lower than none ($p < 0.0001$)
 - tangles have intensity 16.90 lower than ta ($p < 0.0001$)

FIGURE: Splines model plot

```
splinesplot <- ggpredict(splinemod2, terms = c("Dist [all]", "Status", "Tangle")) %>%
  as_tibble() %>%
  rename(Dist = x,
         Status = group,
         Tangle = facet) %>%
  mutate(Tangle = factor(Tangle, levels = c("none", "ta", "tangle"))) %>%
  ggplot(aes(Dist, predicted, color = Tangle, fill = Tangle)) +
  facet_wrap(~Status, nrow = 2) +
  geom_ribbon(aes(ymin = conf.low,
                ymax = conf.high),
            alpha = .3,
            color = NA) +
  geom_line(lwd = 1.25) +
  scale_color_manual(values = c("darkorchid", "darkcyan", "gold")) +
```

```
scale_fill_manual(values = c("darkorchid", "darkcyan", "gold")) +
labs(x = "AIS Length (Microns)",
     y = "Predicted AIS Concentration", color = "") +
theme(legend.position = "none",
      axis.title = element_text(size = 16),
      axis.text = element_text(size = 12)) +
coord_cartesian(ylim = c(125,300), xlim = c(0, 7.5))

splinesplot
```



```
ggsave(splinesplot, filename = "Figures/human_splines.png", width = 6, height = 4)
```

Question 3 - Mean Intensity

- Does mean AIS intensity at each distance differ between AD and non-AD neurons? Is the mean AIS intensity impacted by the presence of a tangle?

Create average intensity dataset. For each neuron, what is the average intensity across the whole distance that was measured

```
avgint <- alldat %>%
  group_by(Status, Brain, Neuron, Tangle) %>%
  summarize(Mean_Intensity = mean(TRIM46_intensity),
```

```

      n = n()) %>%
ungroup()

```

`summarise()` has grouped output by 'Status', 'Brain', 'Neuron'. You can
override using the `.groups` argument.

```
avgint
```

```

## # A tibble: 1,160 x 6
##   Status Brain  Neuron   Tangle Mean_Intensity     n
##   <fct> <chr>  <chr>   <chr>         <dbl> <int>
## 1 Non-AD A14-13 Neuron 1  none          145.    26
## 2 Non-AD A14-13 Neuron 10 none          196.    20
## 3 Non-AD A14-13 Neuron 11 none          197.    26
## 4 Non-AD A14-13 Neuron 12 none          151.    12
## 5 Non-AD A14-13 Neuron 13 none          206.    38
## 6 Non-AD A14-13 Neuron 14 none          172.    21
## 7 Non-AD A14-13 Neuron 15 none          205.    34
## 8 Non-AD A14-13 Neuron 16 none          176.    24
## 9 Non-AD A14-13 Neuron 17 none          176.    24
## 10 Non-AD A14-13 Neuron 18 none          179.    17
## # ... with 1,150 more rows

```

Model the mean intensity by Status and Tangle

```

avgintmodB <- lmer(Mean_Intensity ~ Status * Tangle + (1|Brain), data = avgint)
summary(avgintmodB)

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Mean_Intensity ~ Status * Tangle + (1 | Brain)
##   Data: avgint
##
## REML criterion at convergence: 11241.8
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -2.9697 -0.6795 -0.0915  0.5197  8.1270
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   Brain    (Intercept)  620.2      24.90
##   Residual                    929.4      30.49
## Number of obs: 1160, groups: Brain, 16
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)    211.622      9.034   14.627  23.424 5.26e-13 ***
## StatusAD       -12.023     12.707   14.319  -0.946  0.3598
## Tangleta        6.049      4.126  1145.267   1.466  0.1429

```

```
## Tangletangle          -16.957      11.283 1145.114  -1.503   0.1331
## StatusAD:Tangleta     -11.695       5.595 1153.681  -2.090   0.0368 *
## StatusAD:Tangletangle   7.312      11.945 1145.568   0.612   0.5406
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) SttsAD Tanglt Tngltn StatsAD:Tnglt
## StatusAD      -0.711
## Tangleta      -0.144  0.103
## Tangletangle  -0.047  0.034  0.213
## StatsAD:Tnglt  0.106 -0.107 -0.738 -0.157
## SttsAD:Tngltn  0.045 -0.047 -0.201 -0.945  0.218
```

Ignoring tangle, what is the average difference in mean intensity between AD and non-AD?

```
pairs(emmeans(avgintmodB, specs = "Status"))
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
## contrast      estimate    SE    df t.ratio p.value
## (Non-AD) - AD      13.5 13.2 16.7   1.022  0.3215
##
## Results are averaged over the levels of: Tangle
## Degrees-of-freedom method: kenward-roger
```

No difference

See pairwise comparisons within Status

```
pairs(emmeans(avgintmodB, specs = "Tangle", by = "Status"))
```

```
## Status = Non-AD:
## contrast      estimate    SE    df t.ratio p.value
## none - ta      -6.05  4.14 1145   -1.461  0.3100
## none - tangle   16.96 11.29 1145    1.502  0.2902
## ta - tangle     23.01 11.16 1141    2.062  0.0983
##
## Status = AD:
## contrast      estimate    SE    df t.ratio p.value
## none - ta       5.65  3.78 1149    1.493  0.2945
## none - tangle    9.65  3.93 1149    2.457  0.0377
## ta - tangle     4.00  4.51 1141    0.886  0.6491
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Within neurons with AD, those with tangles have mean intensity 9.65 lower than those without ($p = 0.038$)

See pairwise comparisons within Tangle level

```
pairs(emmeans(avgintmodB, specs = "Status", by = "Tangle"))
```

```
## Tangle = none:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD    12.02 12.7 14.3   0.946  0.3598
##
## Tangle = ta:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD    23.72 13.3 17.3   1.780  0.0927
##
## Tangle = tangle:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD     4.71 17.0 45.7   0.277  0.7832
##
## Degrees-of-freedom method: kenward-roger
```

Within neurons with Tangles, those from AD brains have mean Intensity 4.7 lower than those from non-AD brains ($p = 0.783$)

FIGURE: Mean Intensity by status

Each dot is an average of all of the intensity values for each neuron.

On top of dots are the predicted means and 95% confidence interval from the linear mixed model.

```
# return dataset of predicted mean +/- 95%CI
pred_mean_avgintmod <- ggeffect(avgintmodB, terms = c("Status")) %>%
  as_tibble() %>%
  rename(Status = x)

pred_mean_avgintmod
```

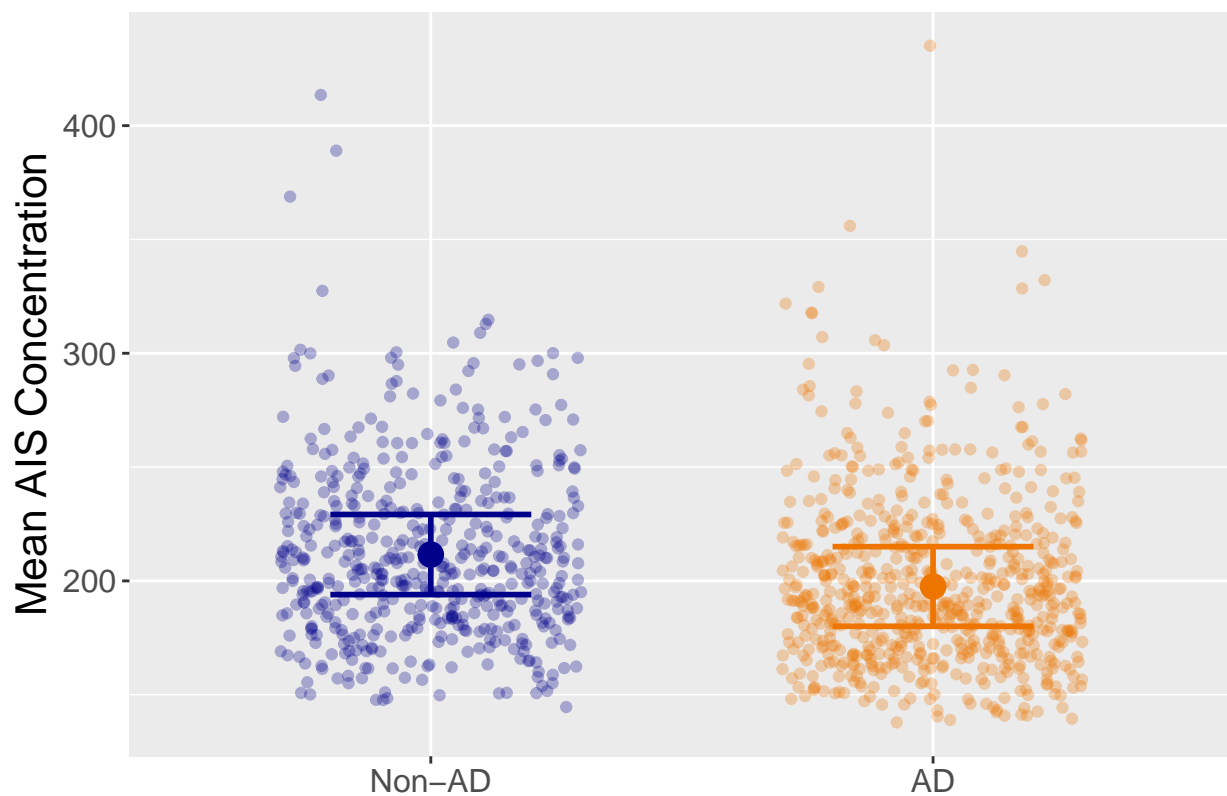
```
## # A tibble: 2 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>    <dbl>    <dbl>    <dbl> <fct>
## 1 Non-AD    212.      8.97     194.     229.   1
## 2 AD       198.      8.91     180.     215.   1
```

```
meanint_supp_plot <- avgint %>%
  ggplot() +
  geom_jitter(aes(x = Status,
                  y = Mean_Intensity,
                  color = Status),
              alpha = .3,
              width = .3) +
  geom_point(data = pred_mean_avgintmod,
             aes(x = Status,
                 y = predicted,
                 color = Status),
             size = 4) +
  geom_errorbar(data = pred_mean_avgintmod,
```

```

aes(x = Status,
    ymin = conf.low,
    ymax = conf.high,
    color = Status),
width = .4,
lwd = 1) +
labs(y = "Mean AIS Concentration", x = "") +
scale_color_manual(values = c("darkblue", "darkorange2")) +
theme(legend.position = "none",
      axis.title = element_text(size = 16),
      axis.text = element_text(size = 12))
meanint_supp_plot

```



```

ggsave(meanint_supp_plot, filename = "Figures/human_meanintensity_status.png", width = 6, height = 4)

```

FIGURE: Mean Intensity by Status and Tangle

Each dot is an average of all of the intensity values for each neuron.

Show jitter plus predicted mean and 95% CI with these colors.

```

# return dataset of predicted mean +/- 95%CI
pred_mean_avgintmod <- ggeffect(avgintmodB, terms = c("Status", "Tangle")) %>%

```

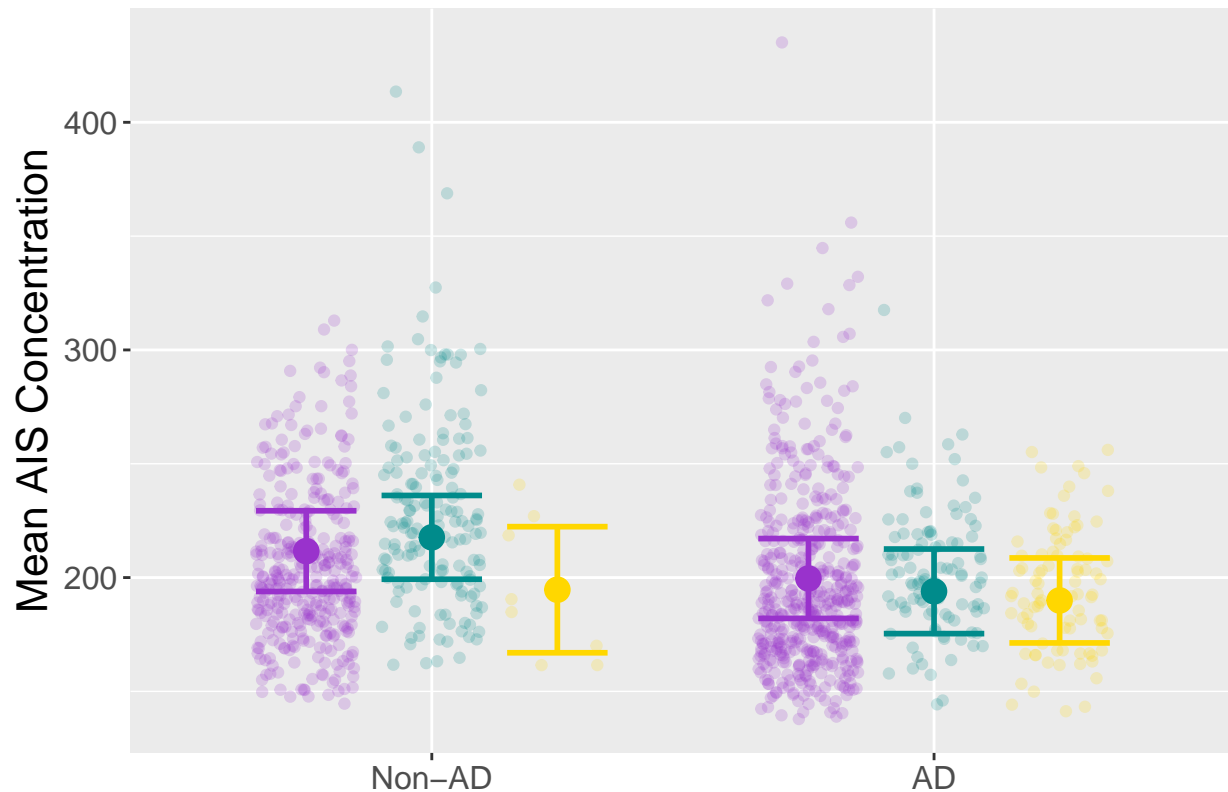
```
as_tibble() %>%
  rename(Status = x)

pred_mean_avgintmod
```

```
## # A tibble: 6 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>      <dbl>   <dbl>   <dbl> <fct>
## 1 Non-AD      212.        9.03    194.    229. none
## 2 AD          200.        8.94    182.    217. none
## 3 Non-AD      218.        9.37    199.    236. ta
## 4 AD          194.        9.47    175.    213. ta
## 5 Non-AD      195.       14.1    167.    222. tangle
## 6 AD          190.        9.51    171.    209. tangle
```

```
meanintplot <- avgint %>%
  ggplot() +
  geom_point(aes(x = Status,
                 y = Mean_Intensity,
                 color = Tangle),
             position = position_jitterdodge(jitter.width = .2),
             alpha = .2) +
  geom_point(data = pred_mean_avgintmod,
             aes(x = Status,
                 y = predicted,
                 color = group),
             position = position_dodge(width = 0.75),
             size = 4) +
  geom_errorbar(data = pred_mean_avgintmod,
               aes(x = Status,
                   ymin = conf.low,
                   ymax = conf.high,
                   color = group),
               position = position_dodge(width = 0.75),
               width = .6,
               lwd = 1) +
  labs(y = "Mean AIS Concentration", x = "") +
  scale_color_manual(values = c("darkorchid", "darkcyan", "gold")) +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12))

meanintplot
```



```
ggsave(meanintplot, filename = "Figures/human_meanintensity_status_tangle.png", width = 6, height = 4)
```

Question 4 & 5 - Max and Min Intensity

Create max and min intensity variables

I tried the **raw** maximum and minimum intensities and determined that a smoothing function works better to establish the true max or min TRIM46 intensity across the distance of each neuron.

Therefore, I define the maximum and minimum intensity based on the rolling average Intensity for each neuron

```
exint <- alldat %>%
  group_by(Brain, Neuron) %>%
  summarise(Max_intensity = max(roll_Intensity, na.rm = TRUE),
            Min_intensity = min(roll_Intensity, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'Brain'. You can override using the
## `.groups` argument.
```

Join the max and min intensity onto the alldat


```
alldat <- alldat %>%
  left_join(exint, by = c("Brain", "Neuron"))
```

Max intensity

Create dataset of just the max value for each neuron.

Check that each neuron only has one maximum

```
alldat %>%
  filter(Max_intensity == roll_Intensity) %>%
  count(Brain, Neuron) %>%
  arrange(-n)
```

```
## # A tibble: 1,160 x 3
##   Brain   Neuron      n
##   <chr>   <chr>   <int>
## 1 A16-174 Neuron 62 (ta)    2
## 2 A11-30  Neuron 1 (tang)    1
## 3 A11-30  Neuron 10    1
## 4 A11-30  Neuron 100    1
## 5 A11-30  Neuron 101    1
## 6 A11-30  Neuron 102 (ta)    1
## 7 A11-30  Neuron 103 (tang)  1
## 8 A11-30  Neuron 104 (ta)    1
## 9 A11-30  Neuron 105 (ta)    1
## 10 A11-30  Neuron 106    1
## # ... with 1,150 more rows
```

Hmm, one neuron has 2 maximums. Look at this neuron

```
alldat %>%
  filter(Brain == "A16-174" & Neuron == "Neuron 62 (ta)")
```

```
## # A tibble: 19 x 9
##   Dist Status Brain   Neuron   TRIM46_intensity Tangle roll_Intensity
##   <dbl> <fct>   <chr>   <chr>         <dbl> <chr>         <dbl>
## 1 0      Non-AD A16-174 Neuron 62 (ta)    242  ta            NA
## 2 0.161 Non-AD A16-174 Neuron 62 (ta)    173  ta            203.
## 3 0.322 Non-AD A16-174 Neuron 62 (ta)    195.  ta            192.
## 4 0.483 Non-AD A16-174 Neuron 62 (ta)    209  ta            208.
## 5 0.644 Non-AD A16-174 Neuron 62 (ta)    220  ta            232.
## 6 0.805 Non-AD A16-174 Neuron 62 (ta)    266  ta            261.
## 7 0.966 Non-AD A16-174 Neuron 62 (ta)    296  ta            290
## 8 1.13   Non-AD A16-174 Neuron 62 (ta)    308  ta            325.
## 9 1.29   Non-AD A16-174 Neuron 62 (ta)    370.  ta            350.
## 10 1.45   Non-AD A16-174 Neuron 62 (ta)    372  ta            379
## 11 1.61   Non-AD A16-174 Neuron 62 (ta)    395.  ta            379
## 12 1.77   Non-AD A16-174 Neuron 62 (ta)    370.  ta            360
## 13 1.93   Non-AD A16-174 Neuron 62 (ta)    315  ta            331.
## 14 2.09   Non-AD A16-174 Neuron 62 (ta)    307.  ta            299.
## 15 2.25   Non-AD A16-174 Neuron 62 (ta)    274  ta            268.
```

```
## 16 2.42 Non-AD A16-174 Neuron 62 (ta) 222 ta 239.
## 17 2.58 Non-AD A16-174 Neuron 62 (ta) 222. ta 210.
## 18 2.74 Non-AD A16-174 Neuron 62 (ta) 187 ta 207.
## 19 2.90 Non-AD A16-174 Neuron 62 (ta) 211 ta NA
## # ... with 2 more variables: Max_intensity <dbl>, Min_intensity <dbl>
```

Just preserve one of these datapoints, so that this neuron does not count twice

```
maxint <- alldat %>%
  filter(Max_intensity == roll_Intensity) %>%
  filter(!(Brain == "A16-174" & Neuron == "Neuron 62 (ta)" & TRIM46_intensity == 372))
maxint
```

```
## # A tibble: 1,160 x 9
##   Dist Status Brain   Neuron TRIM46_intensity Tangle roll_Intensity
##   <dbl> <fct> <chr>   <chr>          <dbl> <chr>          <dbl>
## 1 0.644 AD    A11-30 Neuron 1 (tang) 315. tangle 287.
## 2 0.966 AD    A11-30 Neuron 10 293. none 302.
## 3 3.06 AD    A11-30 Neuron 100 258. none 252.
## 4 1.93 AD    A11-30 Neuron 101 243. none 240.
## 5 1.61 AD    A11-30 Neuron 102 (ta) 180. ta 206.
## 6 1.13 AD    A11-30 Neuron 103 (tang) 220. tangle 222.
## 7 0.322 AD    A11-30 Neuron 104 (ta) 184. ta 197.
## 8 1.29 AD    A11-30 Neuron 105 (ta) 289. ta 260.
## 9 0.483 AD    A11-30 Neuron 106 199. none 227.
## 10 1.61 AD    A11-30 Neuron 107 (tang) 187. tangle 194.
## # ... with 1,150 more rows, and 2 more variables: Max_intensity <dbl>,
## #   Min_intensity <dbl>
```

Each person contributes several max_intensity values (one from each neuron), so use linear mixed model

```
maxmod <- lmer(Max_intensity ~ Status*Tangle + (1|Brain), data = maxint)
summary(maxmod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Max_intensity ~ Status * Tangle + (1 | Brain)
## Data: maxint
##
## REML criterion at convergence: 12497.3
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -2.5964 -0.6650 -0.1073  0.4496  8.5948
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   Brain    (Intercept)         892.2    29.87
##   Residual                    2782.5    52.75
## Number of obs: 1160, groups: Brain, 16
##
## Fixed effects:
```

```
##               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)      258.680    11.119   15.402  23.264 2.03e-13 ***
## StatusAD         -17.120    15.559   14.792  -1.100   0.2888
## Tangleta          8.305     7.062  1069.691   1.176   0.2398
## Tangletriangle   -34.151    19.497  1149.027  -1.752   0.0801 .
## StatusAD:Tangleta -17.184     9.613  1131.439  -1.788   0.0741 .
## StatusAD:Tangletriangle 14.407    20.640  1149.715   0.698   0.4853
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) SttsAD Tngltn Tngltn StatsAD:Tngltn
## StatusAD      -0.715
## Tangleta       -0.202  0.144
## Tangletriangle -0.066  0.047  0.209
## StatsAD:Tngltn  0.148 -0.151 -0.735 -0.154
## SttsAD:Tngltn  0.062 -0.066 -0.197 -0.945  0.214
```

- Compared to non-AD neurons, AD neurons have max intensity 17 lower (not sig)

Ignoring tangle, what is the average difference between AD and non-AD

```
pairs(emmeans(maxmod, specs = "Status"))
```

NOTE: Results may be misleading due to involvement in interactions

```
## contrast      estimate    SE    df t.ratio p.value
## (Non-AD) - AD      18 16.7 19.5   1.078  0.2941
##
## Results are averaged over the levels of: Tangle
## Degrees-of-freedom method: kenward-roger
```

See the pairwise differences within Status

```
pairs(emmeans(maxmod, specs = "Tangle", by = "Status"))
```

```
## Status = Non-AD:
## contrast      estimate    SE    df t.ratio p.value
## none - ta      -8.31   7.10  1069  -1.169  0.4719
## none - tangle   34.15  19.51  1149   1.750  0.1871
## ta - tangle     42.46  19.30  1142   2.200  0.0717
##
## Status = AD:
## contrast      estimate    SE    df t.ratio p.value
## none - ta       8.88   6.53  1153   1.359  0.3627
## none - tangle   19.74   6.78  1153   2.912  0.0102
## ta - tangle     10.87   7.81  1143   1.392  0.3454
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

Within AD neurons, those with a tangle have max intensity 19.7 lower than those without ($p = 0.010$)

See the pairwise differences within Tangle level

```
pairs(emmeans(maxmod, specs = "Status", by = "Tangle"))
```

```
## Tangle = none:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD    17.12 15.6 14.7   1.100  0.2890
##
## Tangle = ta:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD    34.30 17.0 20.7   2.015  0.0571
##
## Tangle = tangle:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD     2.71 25.0 93.7   0.108  0.9139
##
## Degrees-of-freedom method: kenward-roger
```

Within neurons with a tangle, those from AD brains have max intensity 2.7 lower than those from non-AD brains ($p = \text{ns}$)

FIGURE: Max intensity by status

For each neuron, we calculated the maximum TRIM46 intensity value based on the 3-dist rolling average. Those maximum values are plotted here (each neuron contributes one datapoint)

On top of dots are the predicted means and 95% confidence interval from the linear mixed model.

```
# return dataset of predicted mean +/- 95%CI
pred_mean_maxmod <- ggeffect(maxmod, terms = c("Status")) %>%
  as_tibble() %>%
  rename(Status = x)

pred_mean_maxmod
```

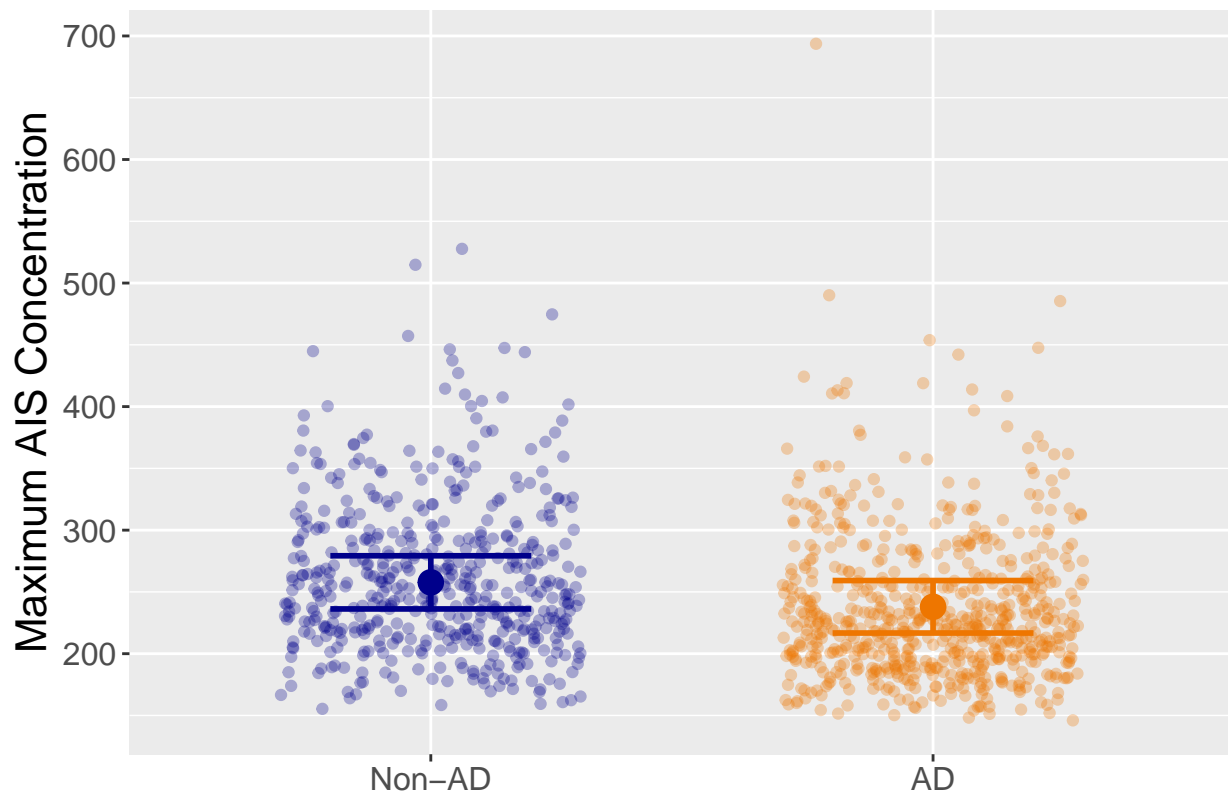
```
## # A tibble: 2 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>    <dbl>   <dbl>   <dbl> <fct>
## 1 Non-AD    258.      11.0    236.    279.  1
## 2 AD       238.      10.8    217.    259.  1
```

```
maxint_supp_plot <- maxint %>%
  ggplot() +
  geom_jitter(aes(x = Status,
                  y = Max_intensity,
                  color = Status),
              alpha = .3,
              width = .3) +
  geom_point(data = pred_mean_maxmod,
             aes(x = Status,
```

```

    y = predicted,
    color = Status),
    size = 4) +
geom_errorbar(data = pred_mean_maxmod,
    aes(x = Status,
        ymin = conf.low,
        ymax = conf.high,
        color = Status),
    width = .4,
    lwd = 1) +
labs(y = "Maximum AIS Concentration", x = "") +
scale_color_manual(values = c("darkblue", "darkorange2")) +
theme(legend.position = "none",
    axis.title = element_text(size = 16),
    axis.text = element_text(size = 12))
maxint_supp_plot

```



```

ggsave(maxint_supp_plot, filename = "Figures/human_maxintensity_status.png", width = 6, height = 4)

```

FIGURE: Max intensity by status and tangle

For each neuron, we calculated the maximum TRIM46 intensity value based on the 3-dist rolling average. Those maximum values are plotted here (each neuron contributes one datapoint)

On top of dots are the predicted means and 95% confidence interval from the linear mixed model.

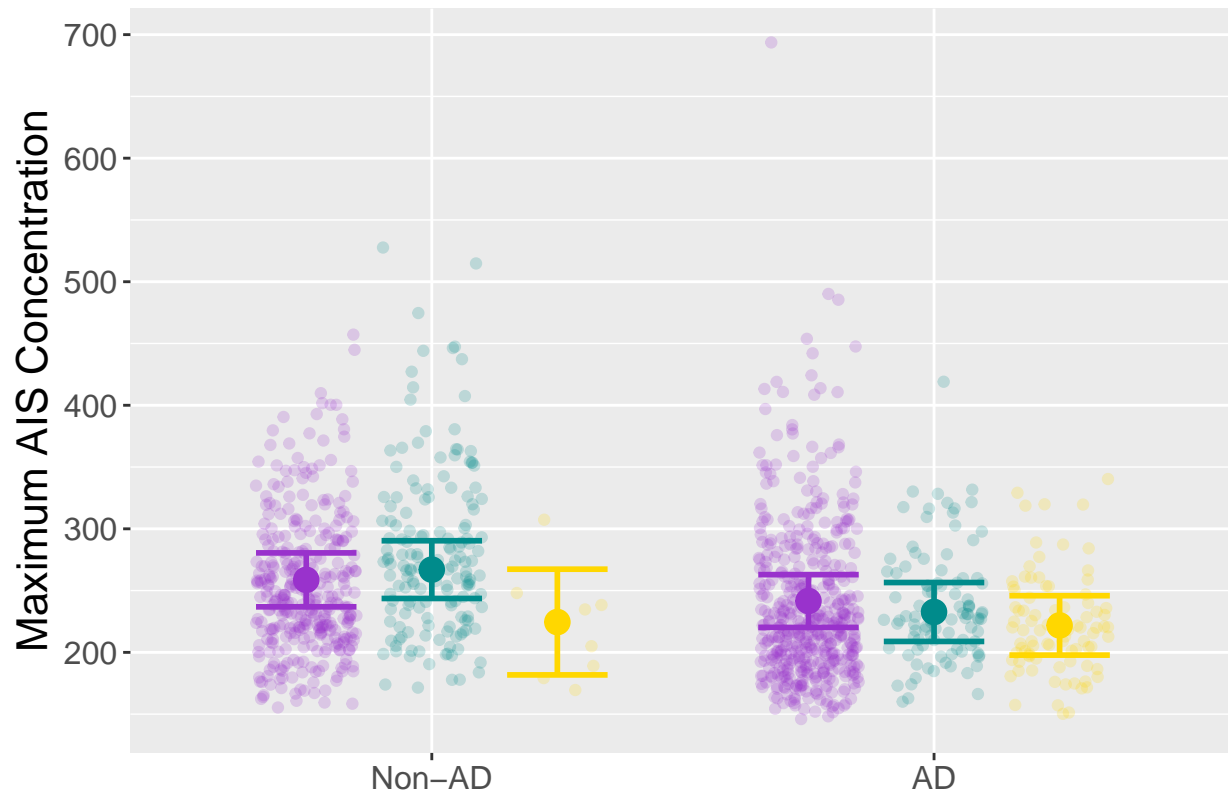
```
# return dataset of predicted mean +/- 95%CI
pred_mean_maxmod <- ggeffect(maxmod, terms = c("Status", "Tangle")) %>%
  as_tibble() %>%
  rename(Status = x)

pred_mean_maxmod
```

```
## # A tibble: 6 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>    <dbl>    <dbl>    <dbl> <fct>
## 1 Non-AD    259.      11.1     237.     280. none
## 2 AD        242.      10.9     220.     263. none
## 3 Non-AD    267.      11.9     244.     290. ta
## 4 AD        233.      12.1     209.     257. ta
## 5 Non-AD    225.      21.8     182.     267. tangle
## 6 AD        222.      12.3     198.     246. tangle
```

```
maxintplot <- maxint %>%
  ggplot() +
  geom_point(aes(x = Status,
                 y = Max_intensity,
                 color = Tangle),
             position = position_jitterdodge(jitter.width = .2),
             alpha = .2) +
  geom_point(data = pred_mean_maxmod,
             aes(x = Status,
                 y = predicted,
                 color = group),
             position = position_dodge(width = .75),
             size = 4) +
  geom_errorbar(data = pred_mean_maxmod,
               aes(x = Status,
                   ymin = conf.low,
                   ymax = conf.high,
                   color = group),
               position = position_dodge(width = .75),
               width = .6,
               lwd = 1) +
  labs(y = "Maximum AIS Concentration", x = "") +
  scale_color_manual(values = c("darkorchid", "darkcyan", "gold")) +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12))

maxintplot
```



```
ggsave(maxintplot, filename = "Figures/human_maxintensity_status_tangle.png", width = 6, height = 4)
```

Min intensity

Create dataset of just the min value for each neuron.

Check that each neuron only has one minimum

```
alldat %>%
  filter(Min_intensity == roll_Intensity) %>%
  count(Brain, Neuron) %>%
  arrange(-n)
```

```
## # A tibble: 1,160 x 3
##   Brain Neuron      n
##   <chr> <chr>   <int>
## 1 A11-30 Neuron 1 (tang) 1
## 2 A11-30 Neuron 10      1
## 3 A11-30 Neuron 100     1
## 4 A11-30 Neuron 101     1
## 5 A11-30 Neuron 102 (ta) 1
## 6 A11-30 Neuron 103 (tang) 1
## 7 A11-30 Neuron 104 (ta) 1
## 8 A11-30 Neuron 105 (ta) 1
## 9 A11-30 Neuron 106     1
```

```
## 10 A11-30 Neuron 107 (tang)      1
## # ... with 1,150 more rows
```

Good

```
minint <- alldat %>%
  filter(Min_intensity == roll_Intensity)
```

Each person contributes several min_intensity values, so use linear mixed model

```
minintmod <- lmer(Min_intensity ~ Status*Tangle + (1|Brain), data = minint)
summary(minintmod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Min_intensity ~ Status * Tangle + (1 | Brain)
## Data: minint
##
## REML criterion at convergence: 10240.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.6089 -0.5979 -0.1335  0.4896  7.3048
##
## Random effects:
## Groups Name Variance Std.Dev.
## Brain (Intercept) 494.9 22.25
## Residual 387.4 19.68
## Number of obs: 1160, groups: Brain, 16
##
## Fixed effects:
##              Estimate Std. Error    df t value Pr(>|t|)
## (Intercept)    171.104     7.973  14.243  21.460 3.01e-12 ***
## StatusAD        -6.330    11.243  14.079  -0.563  0.582
## Tangleta         2.122     2.678 1153.971   0.792  0.428
## Tangletangle     1.355     7.288 1142.810   0.186  0.853
## StatusAD:Tangleta -3.554     3.624 1151.880  -0.981  0.327
## StatusAD:Tangletangle -4.044     7.717 1143.079  -0.524  0.600
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) SttsAD Tanglt Tngltn StatsAD:Tnglt
## StatusAD      -0.709
## Tangleta      -0.106  0.075
## Tangletangle  -0.035  0.025  0.215
## StatsAD:Tnglt  0.078 -0.079 -0.739 -0.159
## SttsAD:Tngltn  0.033 -0.035 -0.203 -0.944  0.220
```

AD neurons have min intensity 6.3 lower than non-AD neurons ($p = ns$)

Pairwise comparisons within Status


```
pairs(emmeans(minintmod, specs = "Tangle", by = "Status"))
```

```
## Status = Non-AD:
## contrast      estimate    SE   df t.ratio p.value
## none - ta      -2.122  2.68 1154  -0.791  0.7086
## none - tangle  -1.355  7.29 1143  -0.186  0.9811
## ta - tangle     0.766  7.20 1141   0.106  0.9938
##
## Status = AD:
## contrast      estimate    SE   df t.ratio p.value
## none - ta       1.433  2.44 1145   0.587  0.8274
## none - tangle    2.689  2.54 1145   1.060  0.5392
## ta - tangle     1.256  2.91 1141   0.431  0.9027
##
## Degrees-of-freedom method: kenward-roger
## P value adjustment: tukey method for comparing a family of 3 estimates
```

No differences

Pairwise comparisons within Tangle levels

```
pairs(emmeans(minintmod, specs = "Status", by = "Tangle"))
```

```
## Tangle = none:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD    6.33 11.2 14.2   0.563  0.5822
##
## Tangle = ta:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD    9.88 11.5 15.7   0.857  0.4046
##
## Tangle = tangle:
## contrast      estimate    SE   df t.ratio p.value
## (Non-AD) - AD   10.37 13.4 28.6   0.773  0.4457
##
## Degrees-of-freedom method: kenward-roger
```

No differences

FIGURE: Min intensity by status

For each neuron, we calculated the minimum TRIM46 intensity value based on the 3-dist rolling average. Those values are plotted here (each neuron contributes one datapoint)

On top of dots are the predicted means and 95% confidence interval from the linear mixed model.

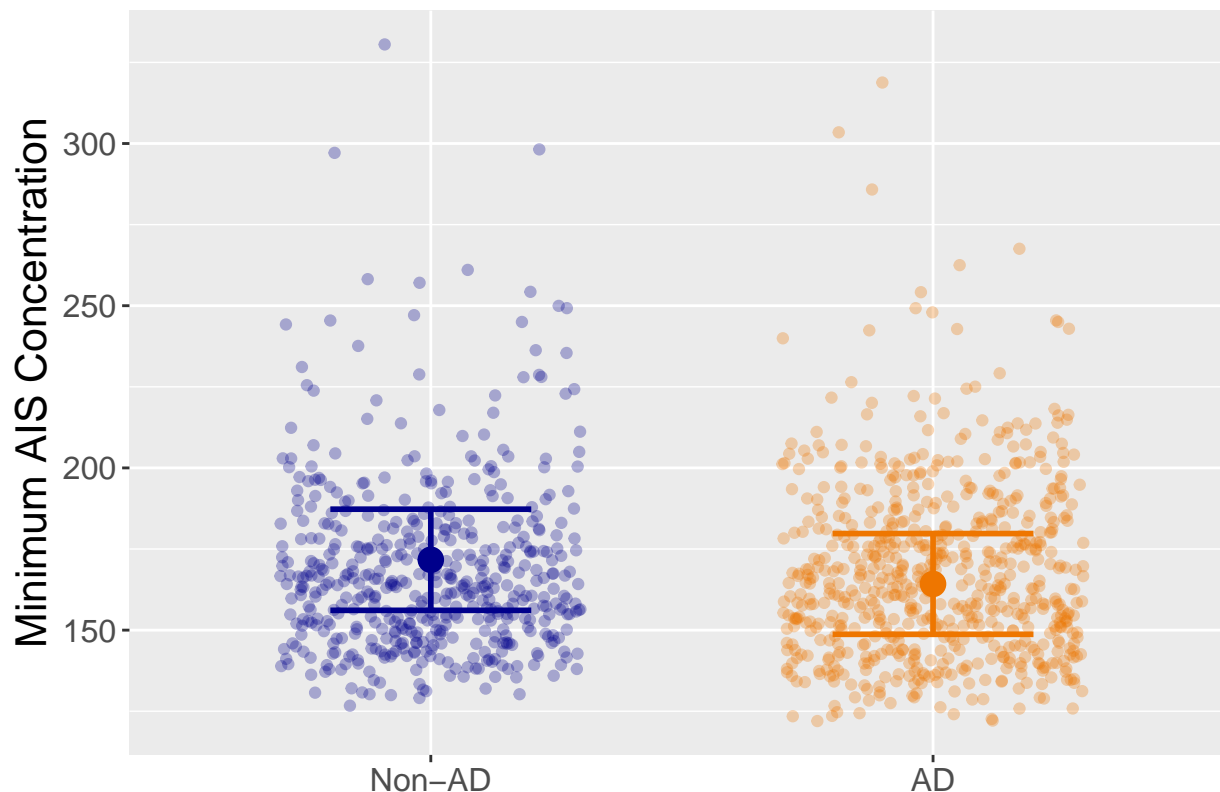
```
# return dataset of predicted mean +/- 95%CI
pred_mean_minmod <- ggeffect(minintmod, terms = c("Status")) %>%
  as_tibble() %>%
  rename(Status = x)

pred_mean_minmod
```

```
## # A tibble: 2 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>      <dbl>    <dbl>    <dbl> <fct>
## 1 Non-AD      172.        7.94     156.     187.  1
## 2 AD          164.        7.92     149.     180.  1
```

```
minint_supp_plot <- minint %>%
  ggplot() +
  geom_jitter(aes(x = Status,
                  y = Min_intensity,
                  color = Status),
              alpha = .3,
              width = .3) +
  geom_point(data = pred_mean_minmod,
             aes(x = Status,
                 y = predicted,
                 color = Status),
             size = 4) +
  geom_errorbar(data = pred_mean_minmod,
               aes(x = Status,
                   ymin = conf.low,
                   ymax = conf.high,
                   color = Status),
               width = .4,
               lwd = 1) +
  labs(y = "Minimum AIS Concentration", x = "") +
  scale_color_manual(values = c("darkblue", "darkorange2")) +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12))

minint_supp_plot
```



```
ggsave(minint_supp_plot, filename = "Figures/human_minintensity_status.png", width = 6, height = 4)
```

FIGURE: Min intensity by status and tangle

For each neuron, we calculated the minimum TRIM46 intensity value based on the 3-dist rolling average. Those values are plotted here (each neuron contributes one datapoint)

On top of dots are the predicted means and 95% confidence interval from the linear mixed model.

```
# return dataset of predicted mean +/- 95%CI
pred_mean_minmod <- ggeffect(minintmod, terms = c("Status", "Tangle")) %>%
  as_tibble() %>%
  rename(Status = x)
```

```
pred_mean_minmod
```

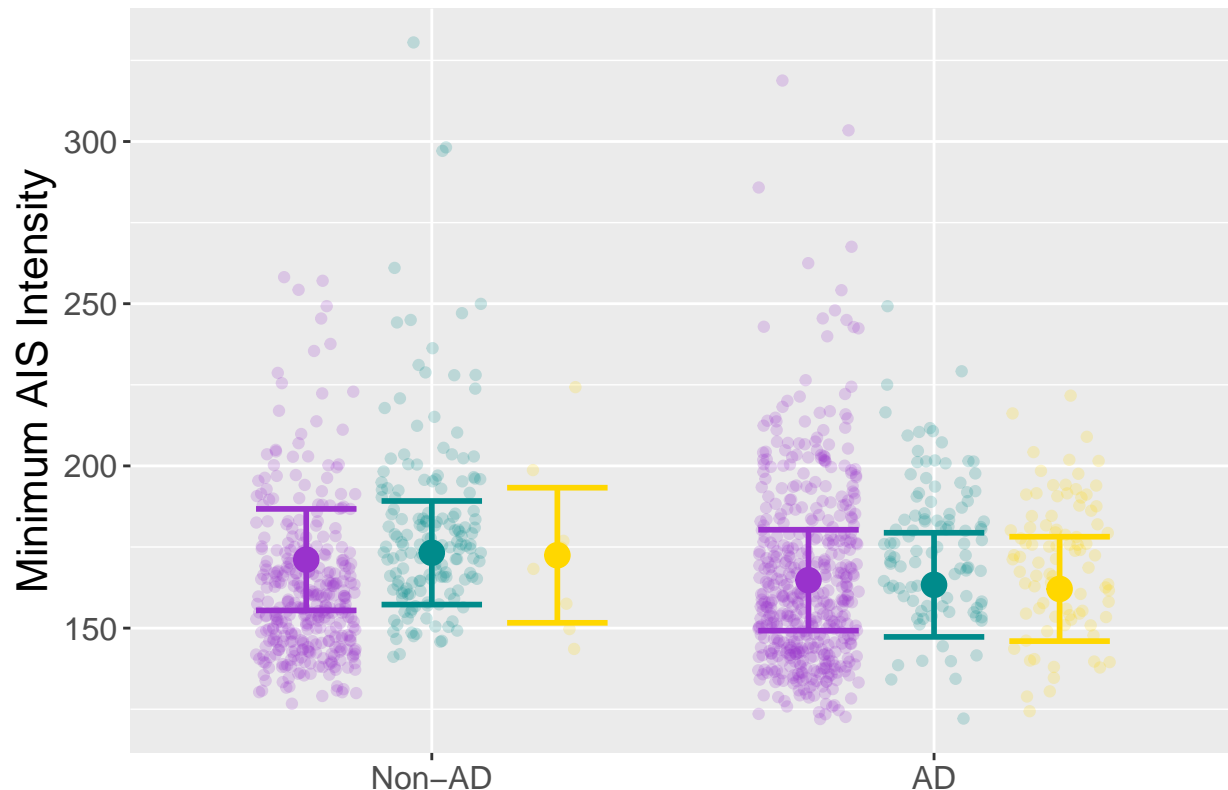
```
## # A tibble: 6 x 6
##   Status predicted std.error conf.low conf.high group
##   <fct>      <dbl>    <dbl>   <dbl>   <dbl> <fct>
## 1 Non-AD    171.     7.97    155.    187. none
## 2 AD        165.     7.93    149.    180. none
## 3 Non-AD    173.     8.14    157.    189. ta
## 4 AD        163.     8.18    147.    179. ta
## 5 Non-AD    172.    10.6    152.    193. tangle
## 6 AD        162.     8.20    146.    178. tangle
```

```

minintplot <- minint %>%
  ggplot() +
  geom_point(aes(x = Status,
                 y = Min_intensity,
                 color = Tangle),
             position = position_jitterdodge(jitter.width = .2),
             alpha = .2) +
  geom_point(data = pred_mean_minmod,
             aes(x = Status,
                 y = predicted,
                 color = group),
             position = position_dodge(width = .75),
             size = 4) +
  geom_errorbar(data = pred_mean_minmod,
               aes(x = Status,
                   ymin = conf.low,
                   ymax = conf.high,
                   color = group),
               position = position_dodge(width = .75),
               width = .6,
               lwd = 1) +
  labs(y = "Minimum AIS Intensity", x = "") +
  scale_color_manual(values = c("darkorchid", "darkcyan", "gold")) +
  theme(legend.position = "none",
        axis.title = element_text(size = 16),
        axis.text = element_text(size = 12))

minintplot

```



```
ggsave(minintplot, filename = "Figures/human_minintensity_status_tangle.png", width = 6, height = 4)
```

Question 6 - AD only Length by severity of disease

6. Within Alzheimer's brains, is there a difference in AIS length across severity of disease?

First create dataset of AIS lengths for only the AD neurons

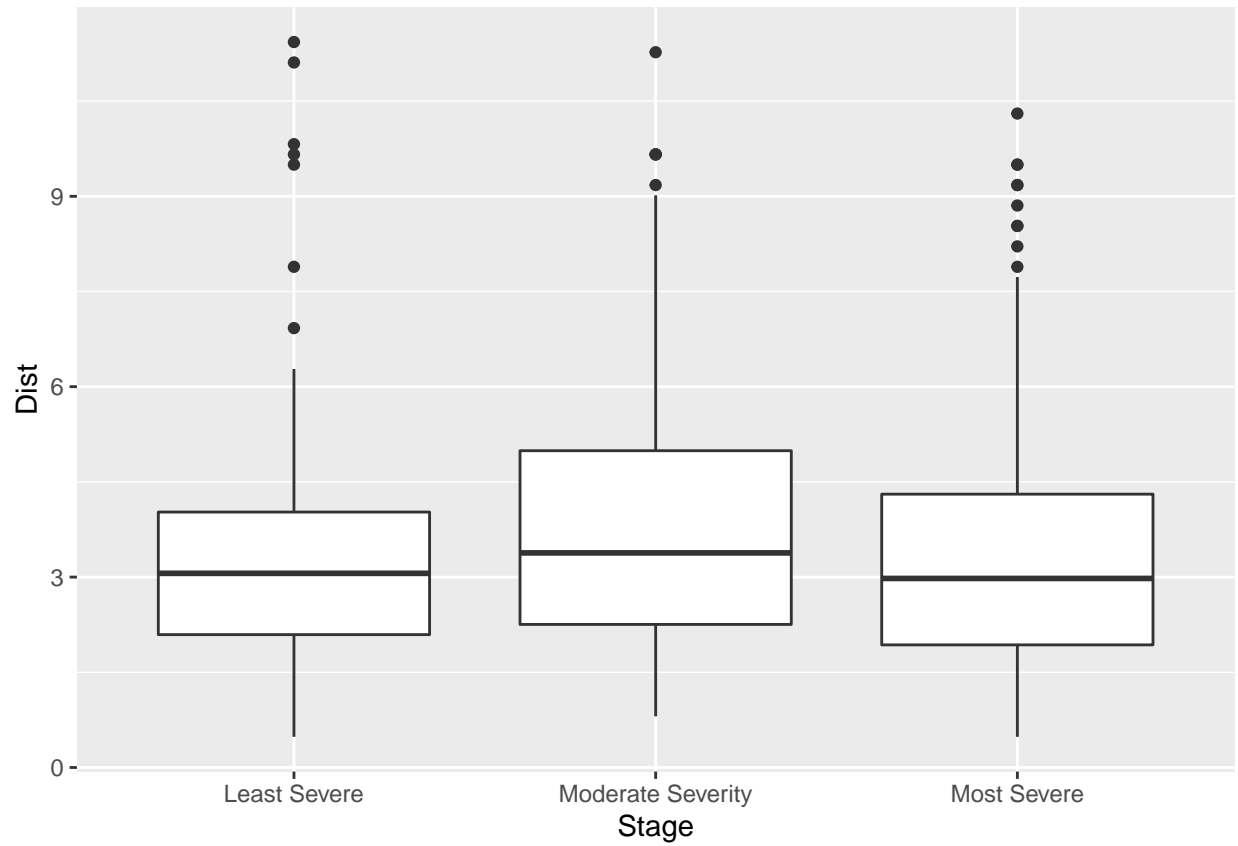
```
ADlength <- maxdist %>%
  filter(Status == "AD")
```

Then add a variable for disease severity: - least severe: AD 1 (A16-109), AD 3 (A15-217) - moderate severity: AD 2 (A15-93), AD 5 (A11-84), AD 7 (A15-167_) - most severe: AD 4 (A13-66), AD 6 (A16-214), AD 8 (A11-30)

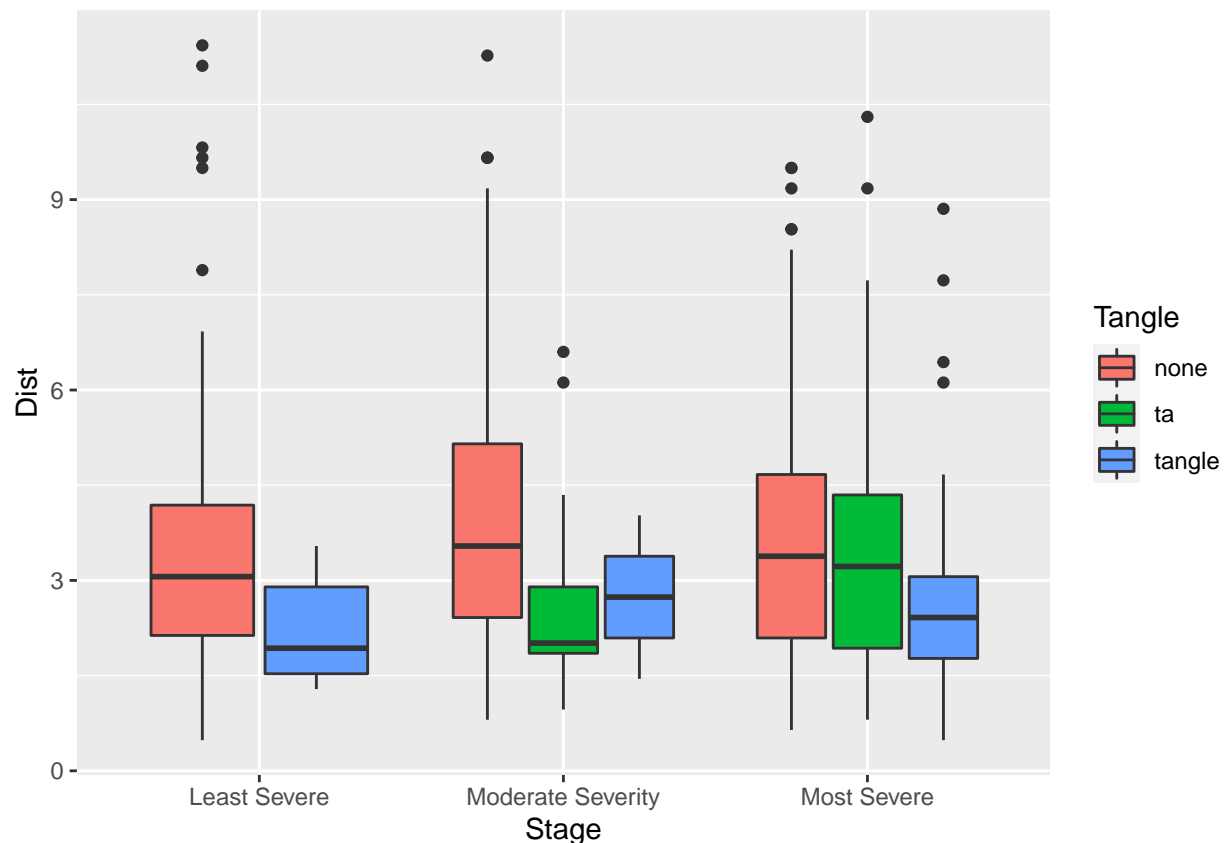
```
ADlength <- ADlength %>%
  mutate(Stage = case_when(Brain == "A16-109" |
    Brain == "A15-217" ~ "Least Severe",
    Brain == "A15-93" |
    Brain == "A11-84" |
    Brain == "A15-167_" ~ "Moderate Severity",
    Brain == "A13-66" |
    Brain == "A16-214" |
    Brain == "A11-30" ~ "Most Severe"))
```

Exploratory plots

```
ADlength %>%  
  ggplot(aes(Stage, Dist)) +  
  geom_boxplot()
```



```
# add tangle  
ADlength %>%  
  ggplot(aes(Stage, Dist, fill = Tangle)) +  
  geom_boxplot()
```



Model with Tangle

```
adlengthmod <- lmer(Dist ~ Stage*Tangle + (1|Brain), data = ADlength)
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```
summary(adlengthmod)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: Dist ~ Stage * Tangle + (1 | Brain)
## Data: ADlength
##
## REML criterion at convergence: 2688
##
## Scaled residuals:
##    Min      1Q  Median      3Q      Max
## -1.8080 -0.7277 -0.1553  0.5029  4.1326
##
## Random effects:
## Groups Name Variance Std.Dev.
## Brain (Intercept) 0.1689 0.411
## Residual 3.4813 1.866
## Number of obs: 657, groups: Brain, 8
##
```

```

## Fixed effects:
##
##              Estimate Std. Error      df t value
## (Intercept)      3.44335    0.33379    4.73345   10.316
## StageModerate Severity      0.48973    0.43544    4.93221    1.125
## StageMost Severe      0.23121    0.44027    5.04623    0.525
## Tangleta      -0.16493    0.25422   641.16528   -0.649
## Tangletangle      -0.94477    0.63192   636.60151   -1.495
## StageModerate Severity:Tangleta      -0.64573    0.62392   645.03257   -1.035
## StageModerate Severity:Tangletangle      0.25935    1.48081   648.92566    0.175
## StageMost Severe:Tangletangle      -0.06322    0.68562   638.62513   -0.092
##
##              Pr(>|t|)
## (Intercept)      0.000201 ***
## StageModerate Severity      0.312462
## StageMost Severe      0.621729
## Tangleta      0.516721
## Tangletangle      0.135392
## StageModerate Severity:Tangleta      0.301079
## StageModerate Severity:Tangletangle      0.861026
## StageMost Severe:Tangletangle      0.926556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) StgMdS StgMsS Tanglt Tngltn StagMdrtSvrty:Tnglt
## StgMdrtSvrt      -0.767
## StageMstSvr      -0.758  0.581
## Tangleta      0.000  0.000 -0.202
## Tangletangl      -0.137  0.105  0.104  0.000
## StagMdrtSvrty:Tnglt  0.000 -0.082  0.082 -0.407  0.000
## StgMdrtSvrty:Tngltn  0.058 -0.079 -0.044  0.000 -0.427  0.045
## StgMstSvr:T      0.126 -0.097 -0.175  0.141 -0.922 -0.057
##
##              StagMdrtSvrty:Tngltn
## StgMdrtSvrt
## StageMstSvr
## Tangleta
## Tangletangl
## StagMdrtSvrty:Tnglt
## StgMdrtSvrty:Tngltn
## StgMstSvr:T      0.393
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient

```

No differences

Question 7 - AD only AIS intensity by severity

7. Within Alzheimer's brains, is there a difference in AIS intensity across severity of disease?
 - a. Splines
 - b. Mean
 - c. Max

Question 7a - Splines for intensity by severity

First create a dataset of all the distances for only AD brains and then add the severity to the dataset.

```
ADdat <- alldat %>%
  filter(Status == "AD") %>%
  mutate(Stage = case_when(Brain == "A16-109" |
    Brain == "A15-217" ~ "Least Severe",
    Brain == "A15-93" |
    Brain == "A11-84" |
    Brain == "A15-167_" ~ "Moderate Severity",
    Brain == "A13-66" |
    Brain == "A16-214" |
    Brain == "A11-30" ~ "Most Severe"))
```

Plot

One line per brain, rolling average averaged over all neurons at a given dist. Then color by Stage.

The rolling average is the average of a 3-dist chunk.

```
rollavgdat <- ADdat %>%
  group_by(Brain, Stage, Dist) %>%
  summarize(Mean_Intensity = mean(roll_Intensity, na.rm = TRUE),
    sd_Intensity = sd(roll_Intensity, na.rm = TRUE),
    n_neurons = n(),
    se_Intensity = sd_Intensity/sqrt(n_neurons))
```

```
## `summarise()` has grouped output by 'Brain', 'Stage'. You can override using
## the ``.groups` argument.
```

```
lineplot <- rollavgdat %>%
  ggplot(aes(Dist, Mean_Intensity)) +
  geom_line(aes(group = Brain, color = Stage)) +
  geom_ribbon(aes(ymin = Mean_Intensity-se_Intensity,
    ymax = Mean_Intensity+se_Intensity,
    group = Brain, fill = Stage), alpha = .3) +
  labs(y = "Rolling Mean AIS Concentration", x = "AIS Length (Microns)") +
  theme(axis.title = element_text(size = 16),
    axis.text = element_text(size = 12)) +
  coord_cartesian(ylim = c(125,300))
```

lineplot

```
## Warning: Removed 16 row(s) containing missing values (geom_path).
```



No modeling needed because there does not seem to be a trend

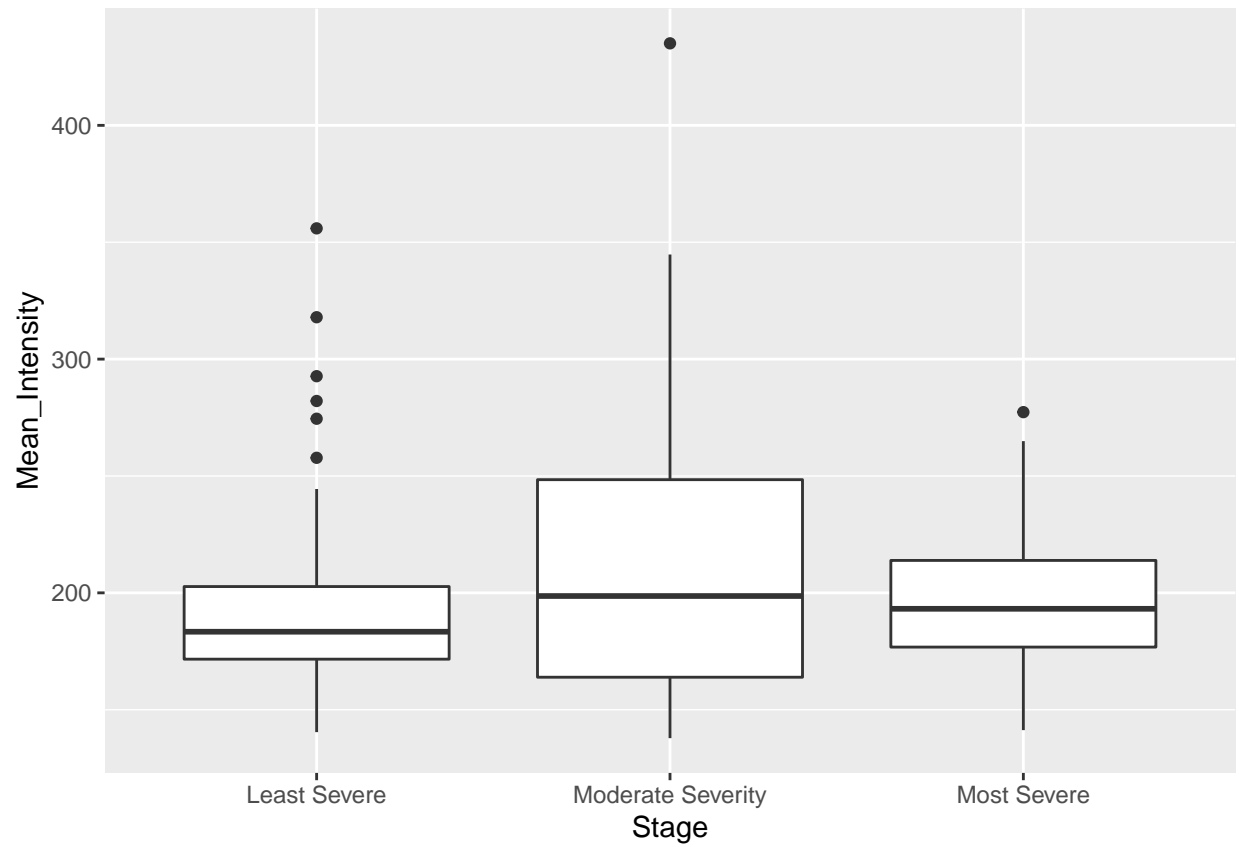
Question 7b - Mean intensity by severity

Start with avgint dataset from above. Filter for AD and add severity variable

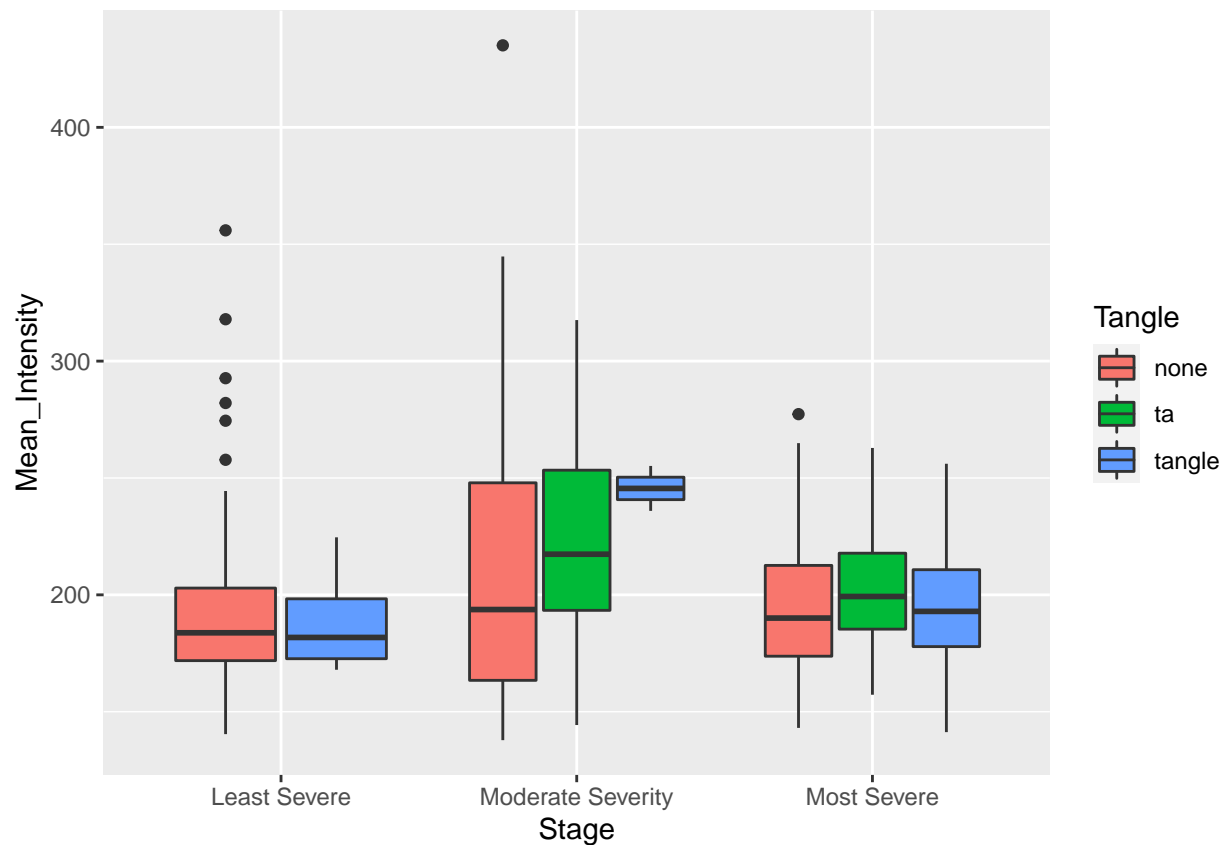
```
ADavg <- avgint %>%
  filter(Status == "AD") %>%
  mutate(Stage = case_when(Brain == "A16-109" |
    Brain == "A15-217" ~ "Least Severe",
    Brain == "A15-93" |
    Brain == "A11-84" |
    Brain == "A15-167_" ~ "Moderate Severity",
    Brain == "A13-66" |
    Brain == "A16-214" |
    Brain == "A11-30" ~ "Most Severe"))
```

Exploratory plot

```
ADavg %>%
  ggplot(aes(Stage, Mean_Intensity)) +
  geom_boxplot()
```



```
# add tangle
ADavg %>%
  ggplot(aes(Stage, Mean_Intensity, fill = Tangle)) +
  geom_boxplot()
```



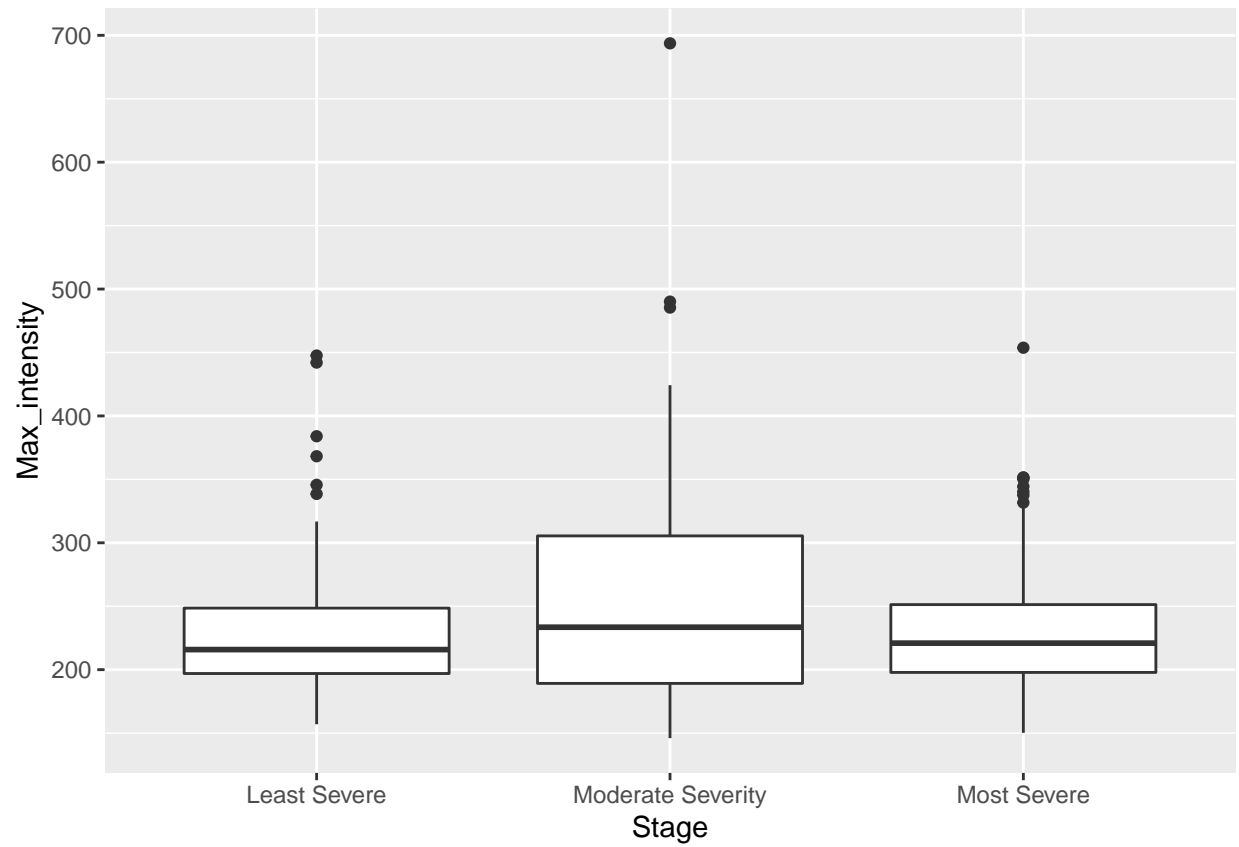
Question 7c - Max intensity by severity

Start with maxint dataset from above. Filter for AD and add severity variable

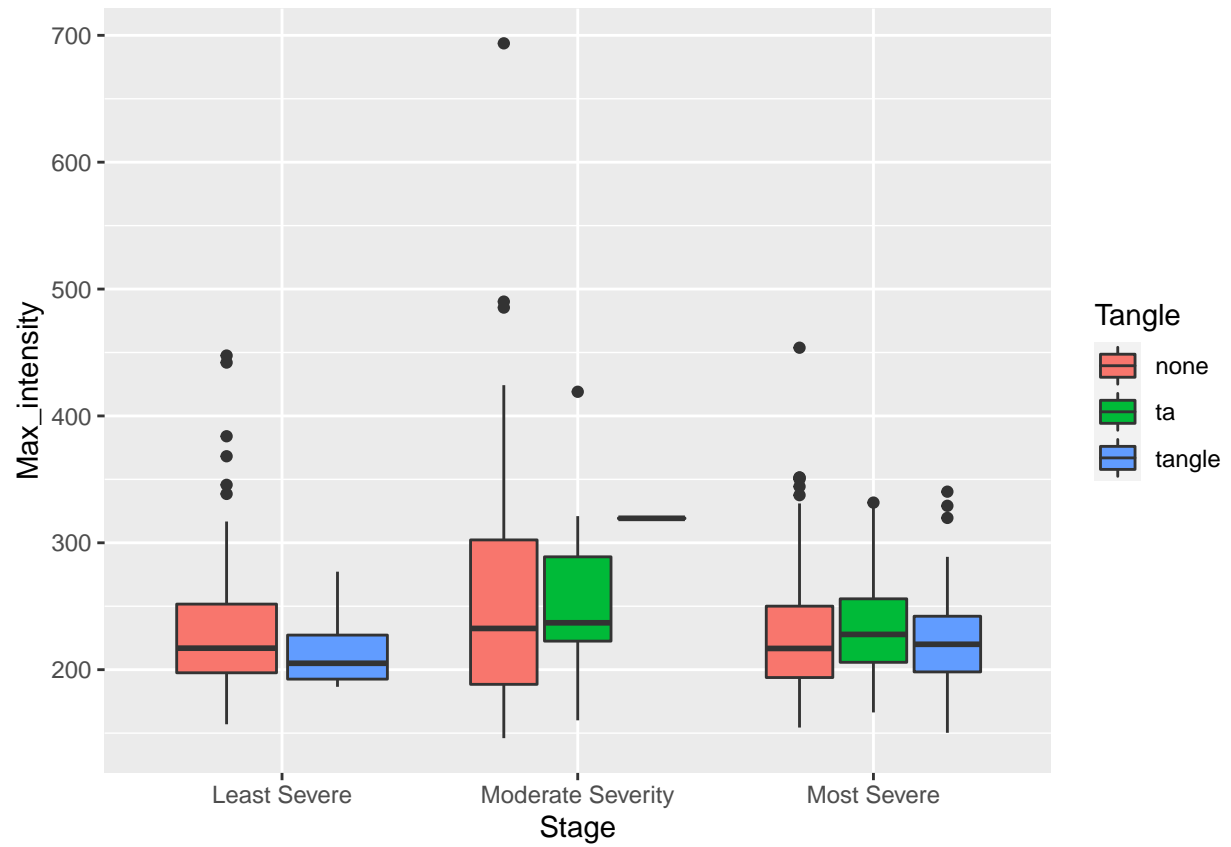
```
ADmax <- maxint %>%
  filter(Status == "AD") %>%
  mutate(Stage = case_when(Brain == "A16-109" |
    Brain == "A15-217" ~ "Least Severe",
    Brain == "A15-93" |
    Brain == "A11-84" |
    Brain == "A15-167_" ~ "Moderate Severity",
    Brain == "A13-66" |
    Brain == "A16-214" |
    Brain == "A11-30" ~ "Most Severe"))
```

Exploratory plot

```
ADmax %>%
  ggplot(aes(Stage, Max_intensity)) +
  geom_boxplot()
```



```
# add tangle
ADmax %>%
  ggplot(aes(Stage, Max_intensity, fill = Tangle)) +
  geom_boxplot()
```



No trend seen here either.