Persistence Experiments

Christian Cole

10/03/2020

Analysis of Persistence Experiments

Go through experiment sets and extract the persistence data. The data are defined by having "ObsType" !=
"Ndata".

Expt 1

```
# Read in ML's data
my.dat = read_xlsx("dat/20191030DatasetPaper.xlsx", sheet = "DataMl")

# only use the persistence experiments for Groups 1 & 2. Groups 3-6 do not
# have enough replicates to be meaningful
persist.dat = my.dat %>% filter(
      Substrate != 'Cott01' &
      Group %in% c(1,2) &
      ObsType != 'Ndata') %>%
  select(c("Substrate", "ObsType", "Count", "Mass", "Duration", "Time", "Group", "Repeat")) %>%
  rename(ObservationType = ObsType,
         TransferTime = Duration,
         PersistenceTime = Time,
         Experiment = Group,
         Replicate = Repeat)

# fix column types
persist.dat$Count = as.numeric(persist.dat$Count)
persist.dat$PersistenceTime = as.numeric(persist.dat$PersistenceTime)
persist.dat$Experiment = as.character(persist.dat$Experiment)
persist.dat$Replicate = as.character(persist.dat$Replicate)
# some errors in the data, fix
persist.dat[persist.dat$Substrate == 'wool01', 'Substrate'] <- 'Wool'
persist.dat[persist.dat$Substrate == 'Wool01', 'Substrate'] <- 'Wool'
persist.dat[persist.dat$Substrate == 'Nylo01', 'Substrate'] <- 'Nylon'
persist.dat[persist.dat$Substrate == 'Wool' & persist.dat$Experiment == '2', 'Substrate'] <- 'Nylon'

# summarise the count data
summ.dat = summarySE(persist.dat, measurevar = "Count", groupvars = c("Substrate", "PersistenceTime", "Experiment")
knitr::kable(head(summ.dat), caption = "Summary of ML's Persistence Data")
```

Table 1: Summary of ML's Persistence Data

Substrate	PersistenceTime	Experiment	N	Count	sd	se	ci
Nylon	0	2	6	49.83333	24.449267	9.981371	25.657932
Nylon	30	2	6	30.33333	22.033308	8.995060	23.122539
Nylon	60	2	6	22.00000	14.615061	5.966574	15.337566

Substrate	PersistenceTime	Experiment	N	Count	sd	se	ci
Nylon	120	2	6	15.00000	14.669697	5.988879	15.394903
Nylon	180	2	6	11.83333	9.907909	4.044887	10.397713
Nylon	360	2	6	3.50000	3.507136	1.431782	3.680513

The data covers a large period of time - potentially weeks - so the time scale needs to be compressed somehow for ease of interpretation. Let's see what we can do.

```
p1 = ggplot(summ.dat, aes(x=PersistenceTime, y = Count, group = Substrate, colour = Substrate, shape=Substrate)) +
  geom_point(position = position_dodge(0.3)) +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08) +
  xlim(c(-5,1500)) +
  scale_y_continuous(breaks = seq(0,60,10)) +
  labs(x = "Persistence Time (min)",
       y = "Particle Count",
       caption = "Error bars: std error") +
  mytheme +
  theme_pubr()

p2 = ggplot(summ.dat, aes(x=PersistenceTime, y = Count, group = Substrate, colour = Substrate, shape=Substrate)) +
  geom_point(position = position_dodge(0.3)) +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08) +
  labs(x = "Persistence Time (min)",
       y = "Particle Count") +
  theme_pubr() +
  theme(axis.title = element_text(size = 8),
        axis.text = element_text(size = 8),
        legend.title = element_blank(),
        legend.position = 'NA')

p1 + annotation_custom(ggplotGrob(p2), xmin = 700, xmax = 1600,
                       ymin = 20, ymax = 65)
```

Error bars: std error
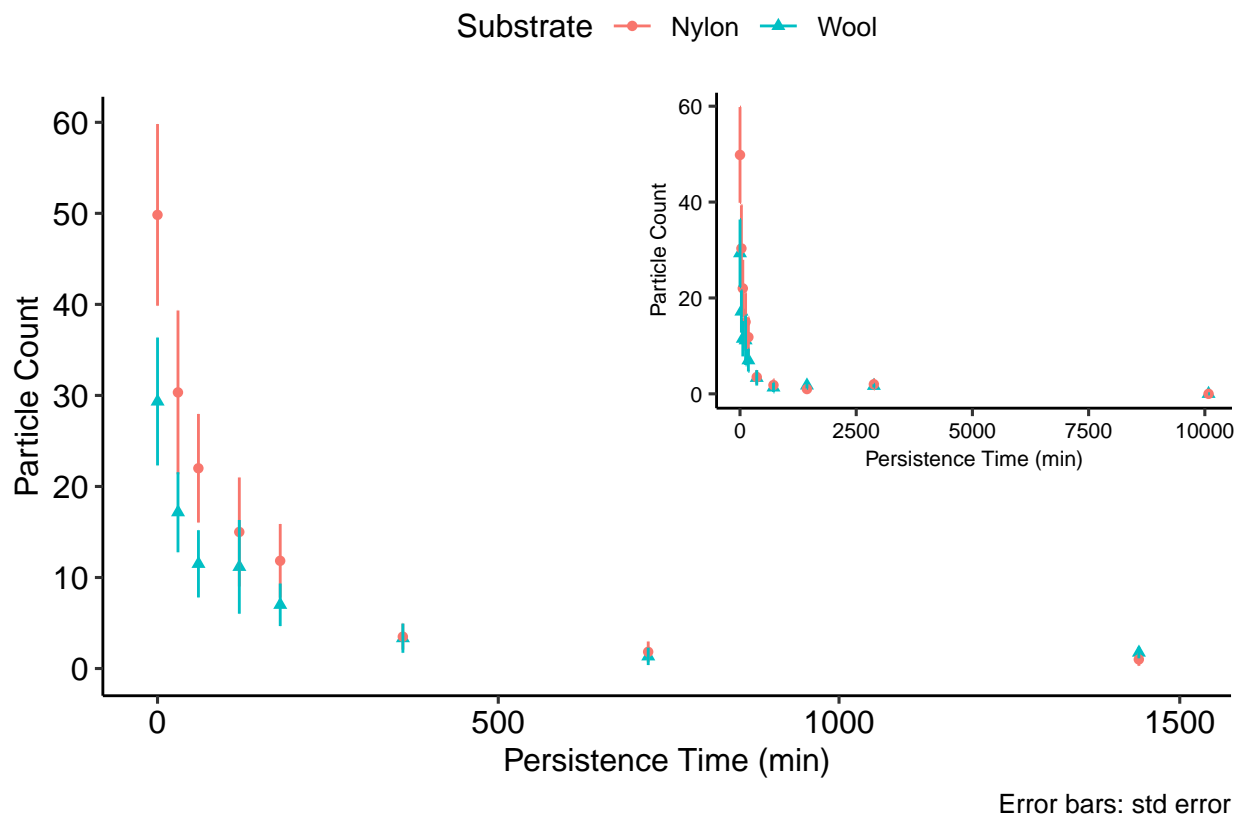
The data seems to be a smooth decay. Let's try some curve-fitting.

This function SSasymp() a self-starting fucntion which attempts to find suitable initial parameters for perform a fit - here a non-linear least squares.

Let's see the Nylon fit first

```
# ideas taken from http://douglas-watson.github.io/post/2018-09_exponential_curve_fitting/
# and https://dataconomy.com/2017/08/nonlinear-least-square-nonlinear-regression-r/
set.seed(12345)
ny.fit = nls(Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha),
             data = summ.dat,
             subset = Substrate == 'Nylon')
ny.fit.err = summ.dat$Count-predict(ny.fit)
ny.nlm_error <- sqrt(mean(ny.fit.err^2))
ny.fit
```

```
## Nonlinear regression model
##   model: Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha)
##    data: summ.dat
##    Countf    Count0 log_alpha
##     2.060    47.063    -4.458
##  residual sum-of-squares: 52.85
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 5.673e-06
```

On first impressions the fit looks fine. The residual error is: 6.42217

A plot of the data and the fit will allow a visual comparison.

```
pts = seq(0,max(summ.dat$PersistenceTime),length=200)
gg.fit = data.frame(y=predict(ny.fit, data.frame(PersistenceTime = pts)), x= pts)

p1 = ggplot(summ.dat[summ.dat$Substrate == 'Nylon',], aes(x = PersistenceTime, y = Count)) +
  labs(title = "Nylon Persistence",
       subtitle = "With non-linear least squares fit",
       x = "Persistence Time (min)",
       y = "Particle Count",
       caption = "Error bars: std error") +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate == 'Nylon',
  geom_line(aes(x = x, y = y), data = gg.fit) +
  xlim(c(-5,1500)) +
  mytheme

p2 = ggplot(summ.dat[summ.dat$Substrate == 'Nylon',], aes(x = PersistenceTime, y = Count)) +
  labs(x = "Persistence Time (min)",
       y = "Particle Count") +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate == 'Nylon',
  geom_line(aes(x = x, y = y), data = gg.fit) +
  inset_theme

p1 + annotation_custom(ggplotGrob(p2), xmin = 700, xmax = 1600,
                       ymin = 20, ymax = 65)
```


Nylon Persistence
With non-linear least squares fit

Error bars: std error
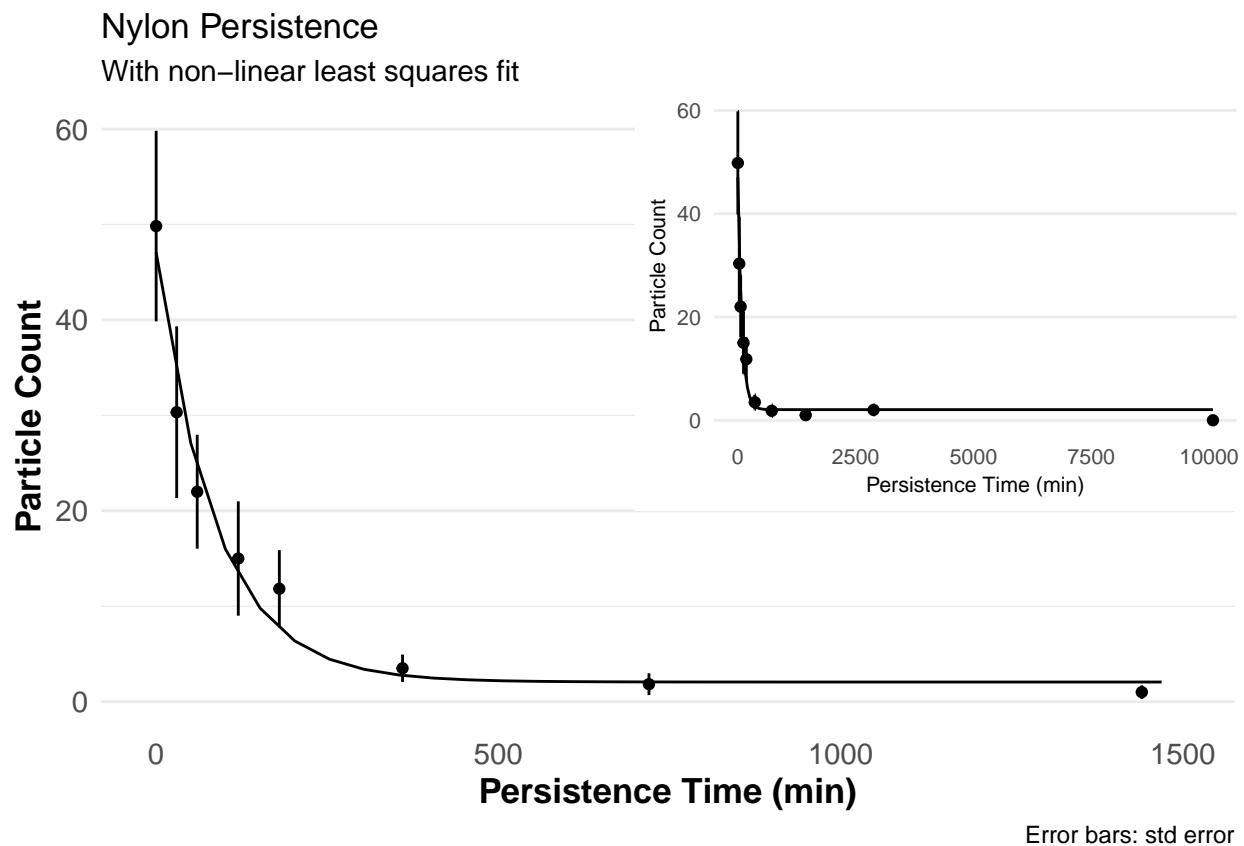
The nylon data looks good, let's to the same with the wool data and compare.

```r
wl.fit = nls(Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha),
             data = summ.dat,
             subset = Substrate == 'Wool')
wl.fit.err = summ.dat$Count-predict(wl.fit)
wl.nlm_error <- sqrt(mean(wl.fit.err^2))
wl.fit
```

```
## Nonlinear regression model
##   model: Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha)
##    data: summ.dat
##    Countf    Count0 log_alpha
##     1.908    27.132    -4.449
##  residual sum-of-squares: 38.19
##
## Number of iterations to convergence: 0
## Achieved convergence tolerance: 4.898e-06
```

Residual error: 6.42218

```r
# generate more points for smooth regression line
pts = seq(0,max(summ.dat$PersistenceTime),length=200)
gg.fit = data.frame(y=predict(wl.fit, data.frame(PersistenceTime = pts)), x= pts)


p1 = ggplot(summ.dat[summ.dat$Substrate == 'Wool',], aes(x = PersistenceTime, y = Count)) +
  labs(title = "Wool Persistence",
       subtitle = "With non-linear least squares fit",
       x = "Persistence Time (min)",
       y = "Particle Count",
       caption = "Error bars: std error") +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate == 'Wool',]
  geom_line(aes(x = x, y = y), data = gg.fit) +
  xlim(c(-5,1500)) +
  mytheme

p2 = ggplot(summ.dat[summ.dat$Substrate == 'Wool',], aes(x = PersistenceTime, y = Count)) +
  geom_point() +
  geom_errorbar(aes(ymin = Count-se, ymax = Count+se), width = 0.08, data = summ.dat[summ.dat$Substrate == 'Wool',]
  geom_line(aes(x = x, y = y), data = gg.fit) +
  inset_theme

p1 + annotation_custom(ggplotGrob(p2), xmin = 700, xmax = 1600,
                       ymin = 10, ymax = 40)
```
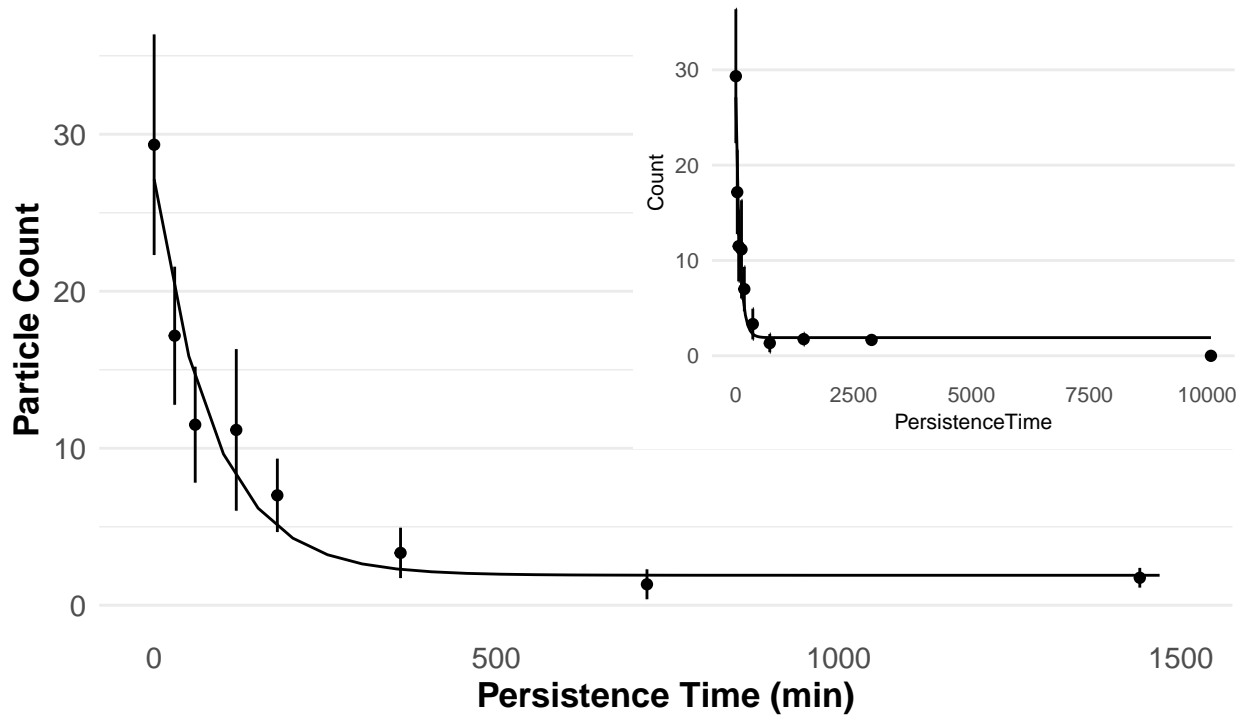
Wool Persistence

With non–linear least squares fit

Error bars: std error

```
cmp.fit.dat = summ.dat %>%
  group_by(Substrate) %>%
  do(fit = nls(Count ~ SSasymp(PersistenceTime, Countf, Count0, log_alpha), data = .)) %>%
  tidy(fit) %>%
  select(Substrate, term, estimate) %>%
  spread(term, estimate) %>%
  mutate(alpha = exp(log_alpha))
knitr::kable(cmp.fit.dat)
```

Substrate	Count0	Countf	log_alpha	alpha
Nylon	47.06339	2.060017	-4.457615	0.0115900
Wool	27.13156	1.908363	-4.449466	0.0116848

Interestingly the decay rate (alpha) is almost the same for the two materials.

Expt 2, 3 & 4

No persistence data.

Expt 5

The only other dataset with persistence results is EM.

```
# Read in Em's data
my.dat = read_xlsx("dat/StrEmCleanDataset.xlsx", sheet = "DataEm", trim_ws = TRUE, col_types = "text")

# Fix the column name for Time - seems to have extra spaces
```

```
my.dat = rename(my.dat, Mass = `Mass (g)`)
my.dat = rename(my.dat, TransferTime = `TransferTime (s)`)
my.dat = rename(my.dat, PersistenceTime = `PersistenceTime (min)`)
#colnames(my.dat)

# only use the Transfer data (PersistenceTime == "0")
# transfer.dat = my.dat %>% filter(PersistenceTime == 0) %>% select(c("Substrate", "ObservationType", "Count", "Mas

# only use expts 7 & 8 as they're the only ones
# with persistence data. Plus for some reason
# the persistence replicates are numbered from 8
persist.dat = my.dat %>% filter(
      Substrate != 'Cott01' &
      Experiment %in% c(7,8) &
      Replicate > 7 &
      ObservationType != 'Ndata') %>%
      select(c("Substrate", "ObservationType", "Count", "Mass", "PersistenceTime", "Experiment", "Replicate", "Note

# fix column types
persist.dat$Count = as.numeric(persist.dat$Count)
persist.dat$PersistenceTime = as.numeric(persist.dat$PersistenceTime)
persist.dat$Experiment = as.character(persist.dat$Experiment)
persist.dat$Replicate = as.character(persist.dat$Replicate)

# summarise the count data
summ.dat = summarySE(persist.dat, measurevar = "Count", groupvars = c("Substrate", "PersistenceTime", "Experiment",
knitr::kable(summ.dat, caption = "Summary of EM's Persistence Data")
```

Table 3: Summary of EM's Persistence Data

| Substrate	PersistenceTime	Experiment	Note	N	Count	sd	se	ci
Nylo01	0	7	C1	2	48.5	4.949747	3.5	44.47172
Nylo01	0	8	C2	2	84.0	16.970563	12.0	152.47446
Nylo01	30	7	C1	2	37.5	4.949747	3.5	44.47172
Nylo01	30	8	C2	2	76.0	7.071068	5.0	63.53102
Nylo01	60	7	C1	2	46.5	21.920310	15.5	196.94617
Nylo01	60	8	C2	2	81.5	2.121320	1.5	19.05931
Nylo01	120	7	C1	1	30.0	NA	NA	NaN
Nylo01	120	8	C2	1	78.0	NA	NA	NaN
Nylo01	180	7	C1	1	14.0	NA	NA	NaN
Nylo01	180	8	C2	1	52.0	NA	NA	NaN
Nylo01	360	7	C1	1	3.0	NA	NA	NaN
Nylo01	360	8	C2	1	8.0	NA	NA	NaN
Nylo01	720	7	C1	1	1.0	NA	NA	NaN
Nylo01	720	8	C2	1	8.0	NA	NA	NaN
Nylo01	1440	7	C1	2	5.5	6.363961	4.5	57.17792
Nylo01	1440	8	C2	2	8.0	11.313709	8.0	101.64964
Nylo01	2880	7	C1	1	2.0	NA	NA	NaN
Nylo01	2880	8	C2	1	3.0	NA	NA	NaN
Nylo01	10080	7	C1	1	0.0	NA	NA	NaN
Nylo01	10080	8	C2	1	0.0	NA	NA	NaN

However, there are nowhere near enough replicates to do anything with the data. Only have 2 reps for 0, 30, 60 and 1440 minutes. All the rest are single-point. #sadface

The difference between experiments '7' and '8' are the camera settings 'C1' and 'C2' which, if they weren't, too different could combine to get more replicates. However, the 'C2' counts are consistently higher than 'C1'

suggesting that they're not the same and cannot be considered replicates.

```
summ.dat = persist.dat %>%
  group_by(Substrate, PersistenceTime, Note) %>%
  summarise(N = n(), Mean = mean(Count), Max = max(Count), Min = min(Count))

ggplot(summ.dat, aes(x=PersistenceTime, y = Mean, colour = Note)) +
  geom_point() +
  geom_linerange(aes(ymin = Min, ymax = Max)) +
  xlim(c(-10, 1500)) +
  labs(title = "Nylon Persistence",
       subtitle = "By different camera settings",
       x = "Persistence Time (min)",
       y = "Particle Count",
       caption = "Error bars: min-max") +
  mytheme
```
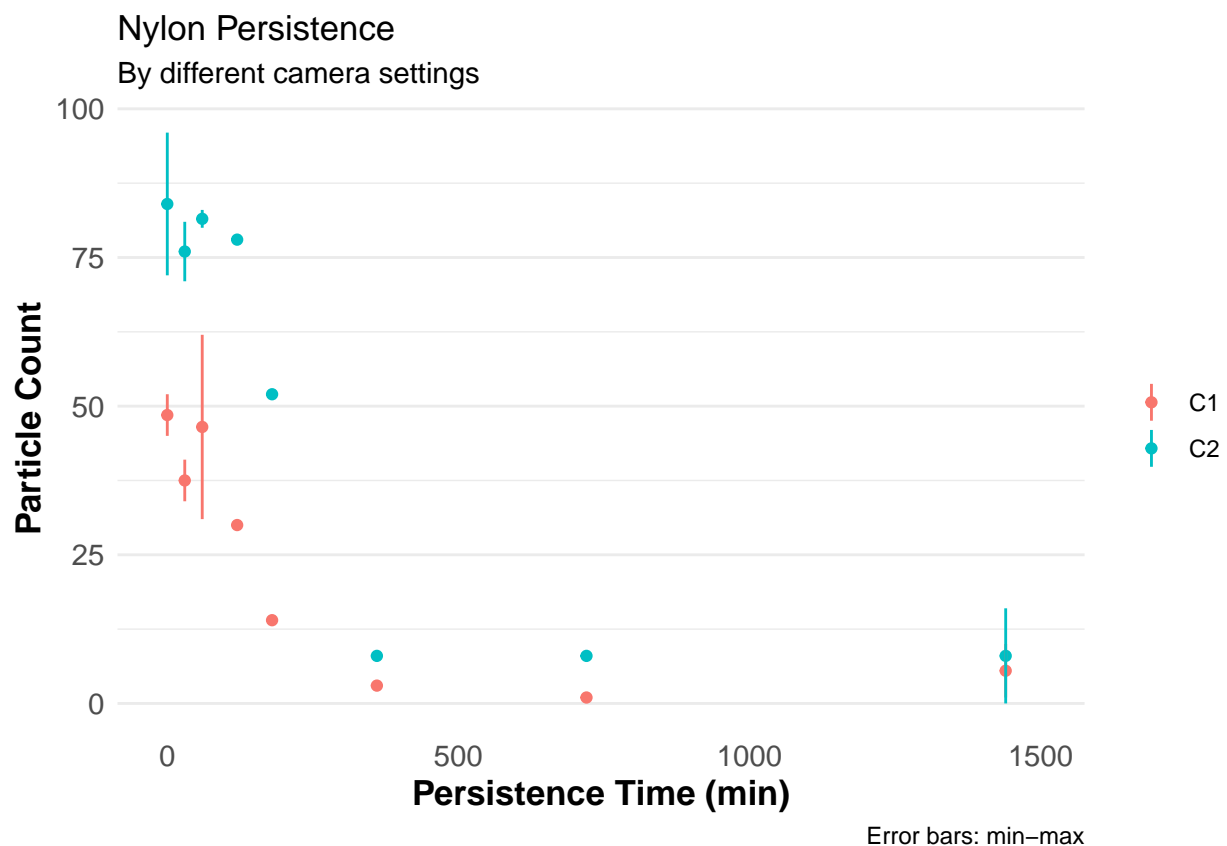

Yup. Data are all over the place. Can't use this in the paper.