Extension Experiments

In this markdown document we'll look at the extensions to the 'baseline' transfer and persistence experiments.

Camera settings data

In this experiment different camera settings were compared. The question being whether different camera settings, labeled 'C1' and 'C2', have an influence on the particle counts.

```
# Read in E data
my.dat = read_xlsx("dat/Counts_data.xlsx", sheet = "E", trim_ws = TRUE, col_types = "text")

# only use the Transfer data (PersistenceTime == "0")
transfer.dat = my.dat %>% filter(`PersistenceTime (min)` == 0) %>%
  select(c("Substrate", "ObservationType", "Count", "Mass (g)", "TransferTime (s)", "Experiment", "Repli
  dplyr::rename(Mass = `Mass (g)`,
         TransferTime = `TransferTime (s)`)

knitr::kable(head(transfer.dat, 20), caption = "'Head' of E Transfer Data")
```

Table 1: 'Head' of E Transfer Data

Substrate	ObservationType	Count	Mass	TransferTime	Experiment	Replicate	Note
Cott01	Ndata	4	0	0	1	1	C1
Nylo01	Ndata	4	0	0	1	1	C1
Cott01	Uvpo1	100	0	0	1	1	C1
Cott01	Uvpo1	81	1000	30	1	1	C1
Nylo01	Uvpo1	21	1000	30	1	1	C1
Cott01	Ndata	0	0	0	1	2	C1
Nylo01	Ndata	2	0	0	1	2	C1
Cott01	Uvpo1	151	0	0	1	2	C1
Cott01	Uvpo1	135	1000	30	1	2	C1
Nylo01	Uvpo1	35	1000	30	1	2	C1
Cott01	Ndata	4	0	0	1	3	C1
Nylo01	Ndata	4	0	0	1	3	C1
Cott01	Uvpo1	46	0	0	1	3	C1
Cott01	Uvpo1	47	1000	30	1	3	C1
Nylo01	Uvpo1	16	1000	30	1	3	C1
Cott01	Ndata	4	0	0	1	4	C1
Nylo01	Ndata	5	0	0	1	4	C1
Cott01	Uvpo1	24	0	0	1	4	C1
Cott01	Uvpo1	21	1000	30	1	4	C1
Nylo01	Uvpo1	5	1000	30	1	4	C1

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

# add a photoID column to keep track of which datapoint is which
photo.id = rep(paste0("P", seq(1:5)), as.integer(nrow(transfer.dat)/5+1))
transfer.dat = data.frame(PhotoID = photo.id[1:nrow(transfer.dat)], transfer.dat)
```

```
# split out each set of photos...
p1 = transfer.dat %>% filter(PhotoID == "P1") %>% select(Mass,TransferTime,Substrate,Experiment,Replica
p2 = transfer.dat %>% filter(PhotoID == "P2") %>% select(Mass,TransferTime,Substrate,Experiment,Replica
p3 = transfer.dat %>% filter(PhotoID == "P3") %>% select(Mass,TransferTime,Substrate,Experiment,Replica
p4 = transfer.dat %>% filter(PhotoID == "P4") %>% select(Mass,TransferTime,Substrate,Experiment,Replica
p5 = transfer.dat %>% filter(PhotoID == "P5") %>% select(Mass,TransferTime,Substrate,Experiment,Replica

# ...and merge
merge.dat = data.frame(p5[,c("Mass","TransferTime","Substrate","Experiment","Replicate", "Note")], p1$C
names(merge.dat) = c("Mass","TransferTime","Substrate","Experiment","Replicate", "Note","P1","P2","P3",

# The above has the side-effect of marking P1-P3 images with the
# Mass and TransferTime of P5. useful for turning into long format

# convert to long format
tlong.dat = gather(merge.dat, Photo, Count, P1:P5)

# rename substrates
tlong.dat[tlong.dat$Substrate == 'Wool01', 'Substrate'] <- 'Wool'
tlong.dat[tlong.dat$Substrate == 'Nylo01', 'Substrate'] <- 'Nylon'
knitr::kable(head(tlong.dat, 12), caption = "'Head' of E data")
```

Table 2: ‘Head’ of E data

Mass	TransferTime	Substrate	Experiment	Replicate	Note	Photo	Count
1000	30	Nylon	1	1	C1	P1	4
1000	30	Nylon	1	2	C1	P1	0
1000	30	Nylon	1	3	C1	P1	4
1000	30	Nylon	1	4	C1	P1	4
1000	30	Nylon	1	6	C1	P1	1
1000	30	Nylon	2	1	C2	P1	3
1000	30	Nylon	2	2	C2	P1	0
1000	30	Nylon	2	3	C2	P1	3
1000	30	Nylon	2	4	C2	P1	3
1000	30	Nylon	2	6	C2	P1	0
1000	120	Nylon	3	1	C1	P1	8
1000	120	Nylon	3	2	C1	P1	4

Mass Data

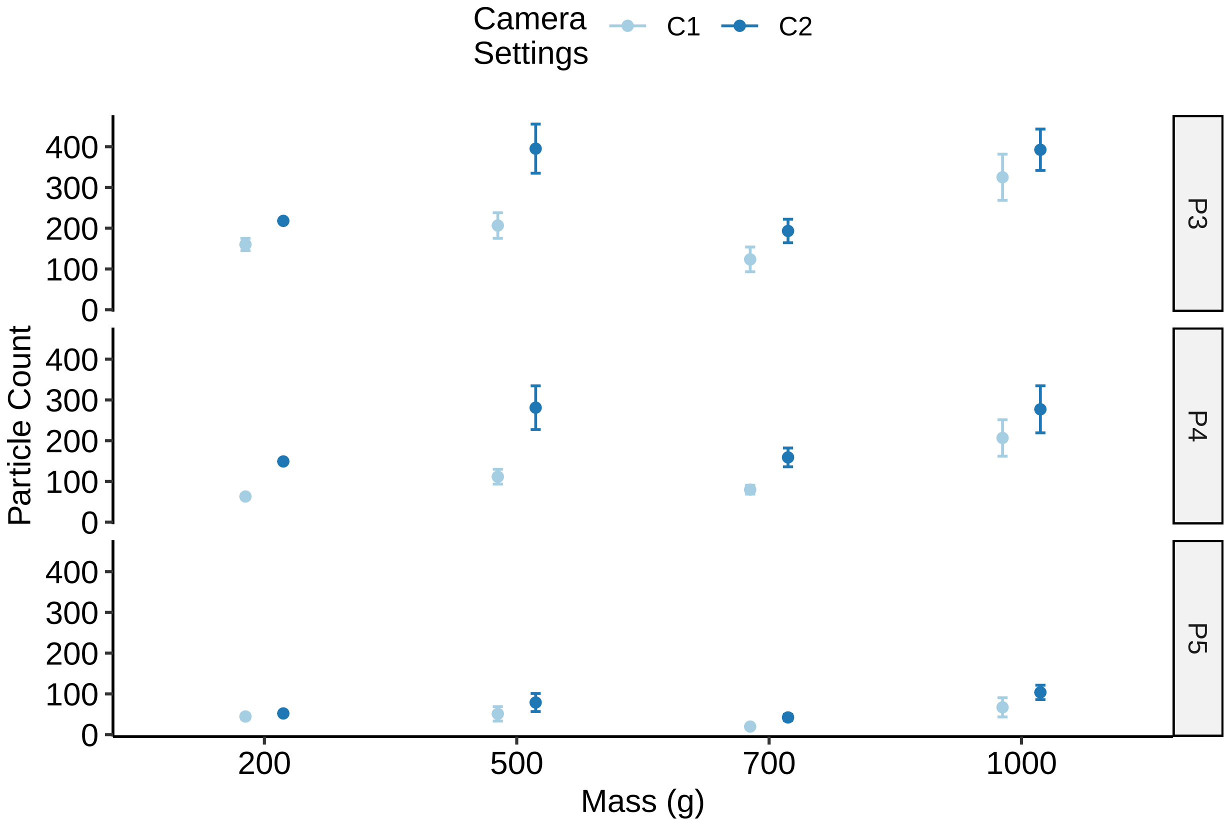
Let’s start with the 60s data over the various masses.

```
# select 60s transfer time
t60.dat = tlong.dat %>%
  filter(TransferTime == 60) %>%
  group_by(Mass, TransferTime, Experiment, Substrate, Note, Photo) %>%
  summarise(N = n(), mean = mean(Count), se = stdErr(Count))
knitr::kable(head(t60.dat), caption = "'Head' of Camera Data")
```

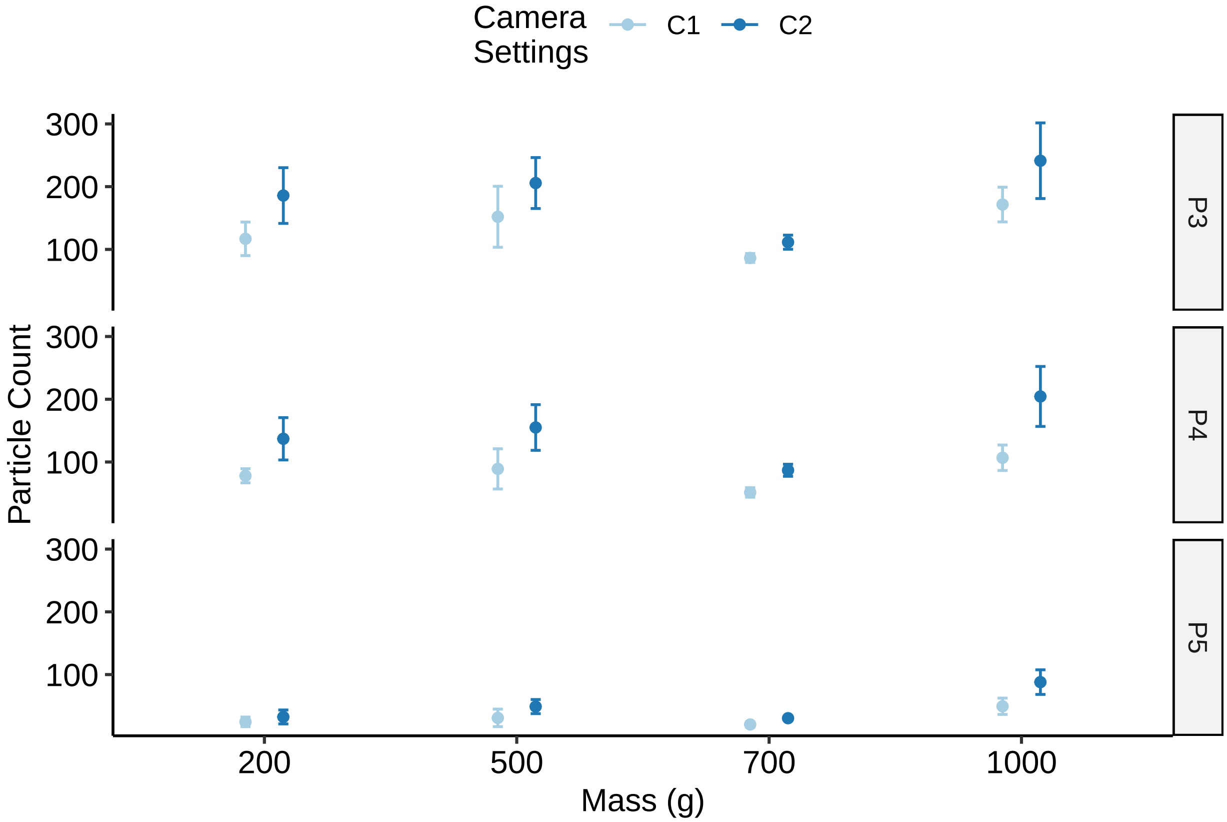
Table 3: ‘Head’ of Camera Data

Mass	TransferTime	Experiment	Substrate	Note	Photo	N	mean	se
1000	60	17	Wool	C1	P1	5	0.2	0.2000000
1000	60	17	Wool	C1	P2	5	1.0	0.3162278
1000	60	17	Wool	C1	P3	5	325.0	56.6277317
1000	60	17	Wool	C1	P4	5	206.6	44.7533239
1000	60	17	Wool	C1	P5	5	67.0	23.5669260
1000	60	18	Wool	C2	P1	6	0.5	0.2236068

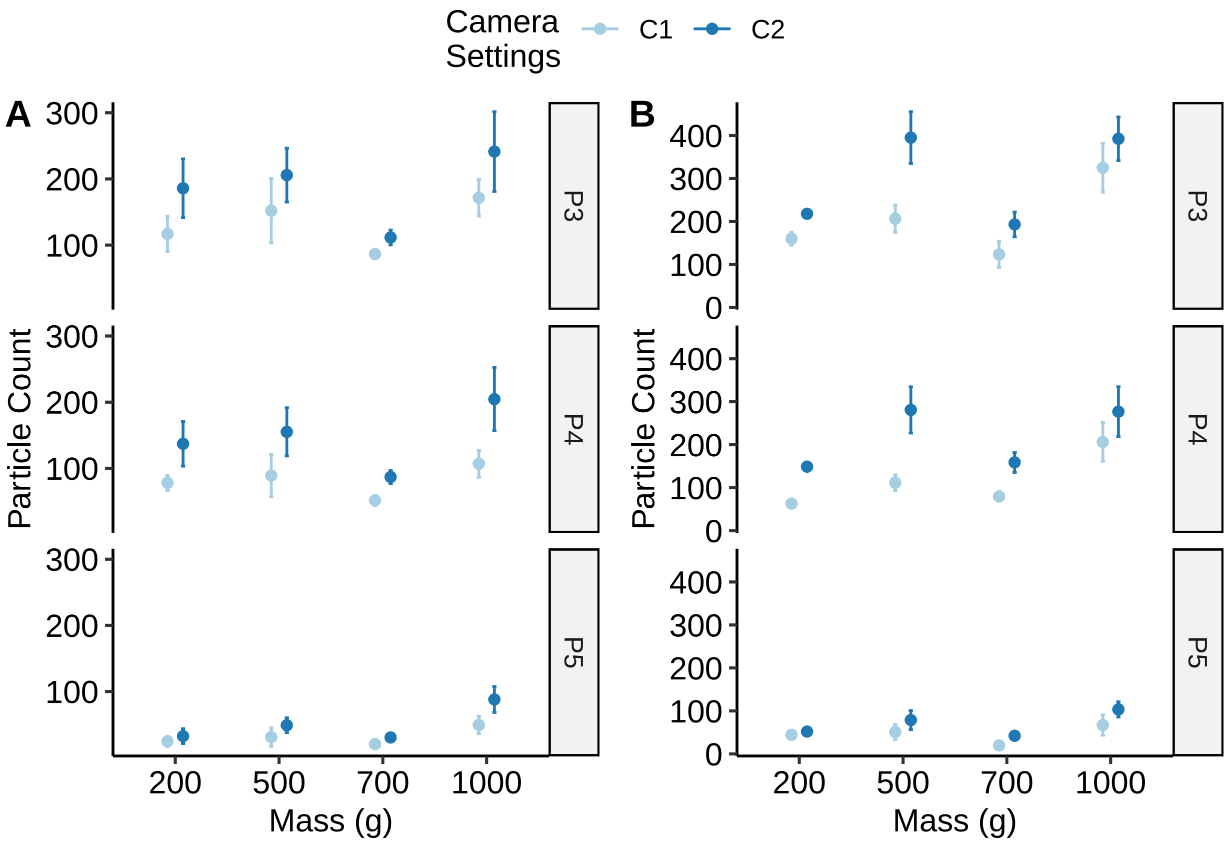
```
t60wl.dat = t60.dat %>% filter(Substrate == "Wool", Photo %in% c('P3', 'P4', 'P5'))

twp = ggplot(t60wl.dat, aes(x = numericFactorOrder(Mass), y = mean, colour = Note)) +
  labs(
       x = "Mass (g)",
       y = "Particle Count",
       colour = "Camera\nSettings") +
  scale_color_brewer(palette = 'Paired') +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.08, position = position_dodge(width
  facet_grid(Photo ~ .) +
  facet_theme +
  theme_pubr()

twp
```



```
t60ny.dat = t60.dat %>% filter(Substrate == "Nylon", Photo %in% c('P3', 'P4', 'P5'))

tnp = ggplot(t60ny.dat, aes(x = numericFactorOrder(Mass), y = mean, colour = Note)) +
  labs(
      x = "Mass (g)",
      y = "Particle Count",
      colour = "Camera\nSettings") +
  scale_color_brewer(palette = 'Paired') +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.08, position = position_dodge(width
  facet_grid(Photo ~ .) +
  facet_theme +
  theme_pubr()

tnp
```



```
nnwt = ggarrange(tnp, twp,
          labels = c('A', 'B'), ncol = 2, common.legend = TRUE,
          legend = 'top')

nnwt
```



```
#ggsave('extcamera1.png', nnwt, height = 4.5, units = 'in')
```

Transfer Time

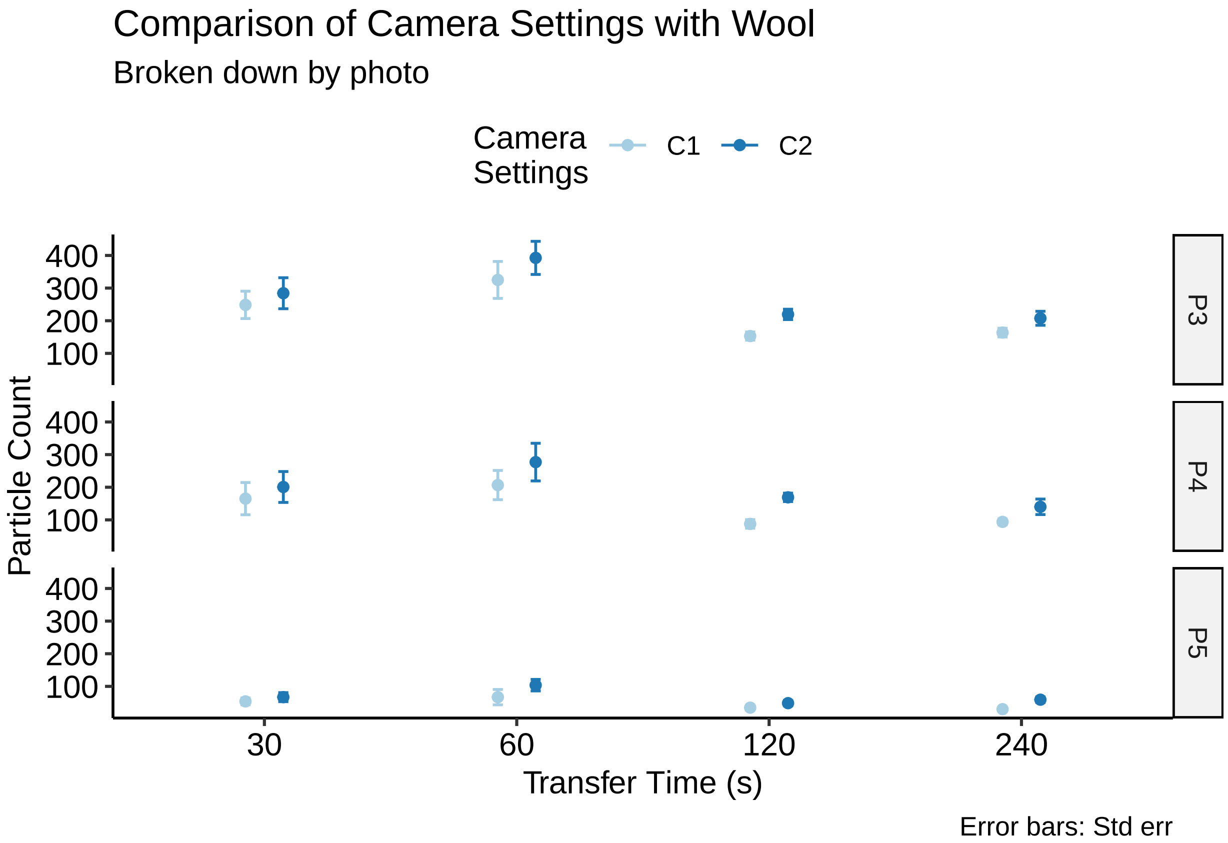
Compare camera settings across Transfer Time.

```
m1000.dat = tlong.dat %>%
  filter(Mass == 1000, TransferTime != 10) %>%
  group_by(Mass, TransferTime, Experiment, Substrate, Note, Photo) %>%
  summarise(N = n(), mean = mean(Count), se = stdErr(Count))

# remove single rep expts
m1000.dat = m1000.dat[m1000.dat$N > 1,]


knitr::kable(head(m1000.dat), caption = "'Head' of Camera Data")
```

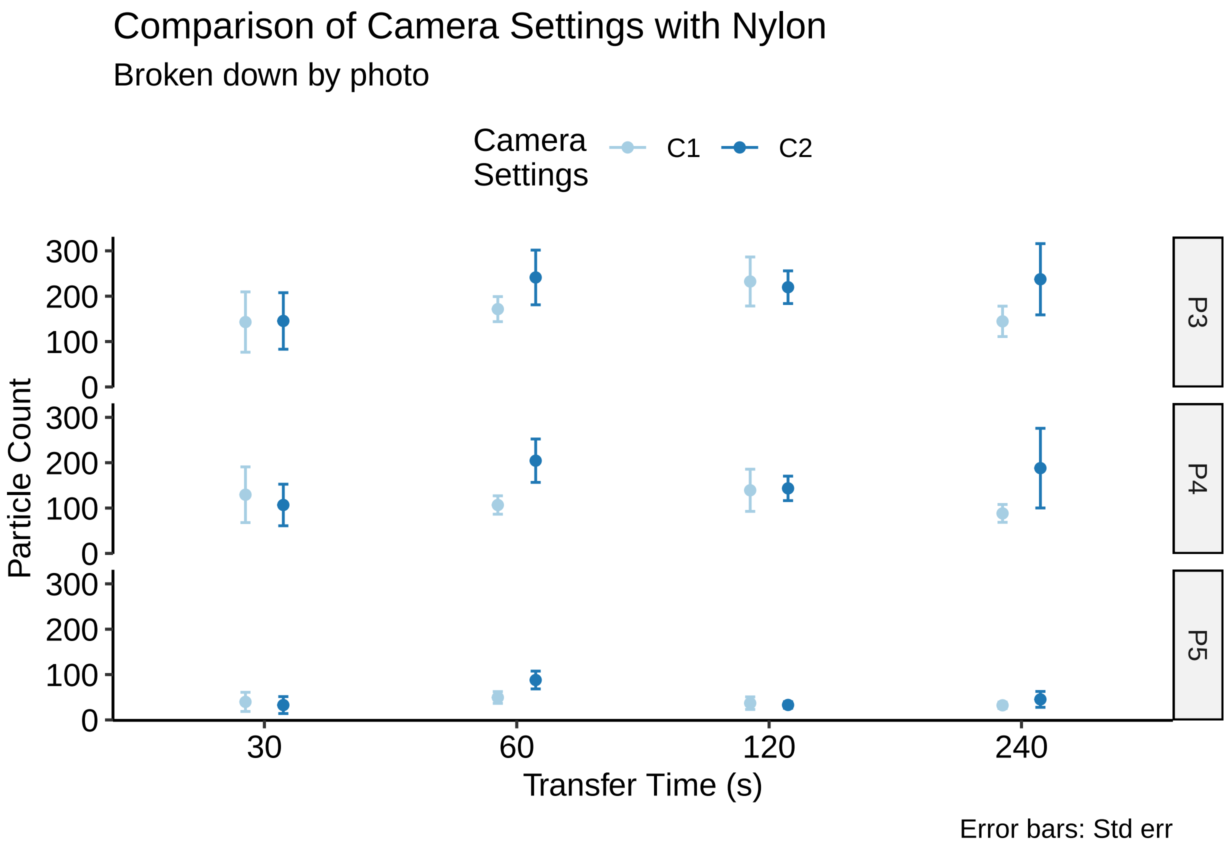
Table 4: ‘Head’ of Camera Data

Mass	TransferTime	Experiment	Substrate	Note	Photo	N	mean	se
1000	120	19	Wool	C1	P1	5	0.0	0.0000000
1000	120	19	Wool	C1	P2	5	1.0	0.4472136
1000	120	19	Wool	C1	P3	5	153.0	12.5499004
1000	120	19	Wool	C1	P4	5	87.6	13.1742172
1000	120	19	Wool	C1	P5	5	34.8	7.1512237
1000	120	20	Wool	C2	P1	5	0.0	0.0000000

```r
m1000wl.dat = m1000.dat %>% filter(Substrate == "Wool", Photo %in% c('P3', 'P4', 'P5'))

ggplot(m1000wl.dat, aes(x = numericFactorOrder(TransferTime), y = mean, colour = Note)) +
  labs(title = "Comparison of Camera Settings with Wool",
       subtitle = "Broken down by photo",
       x = "Transfer Time (s)",
       y = "Particle Count",
       colour = "Camera\nSettings",
       caption = "Error bars: Std err") +
  scale_color_brewer(palette = 'Paired') +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.08, position = position_dodge(width =
  facet_grid(Photo ~ .) +
  facet_theme +
  theme_pubr()
```

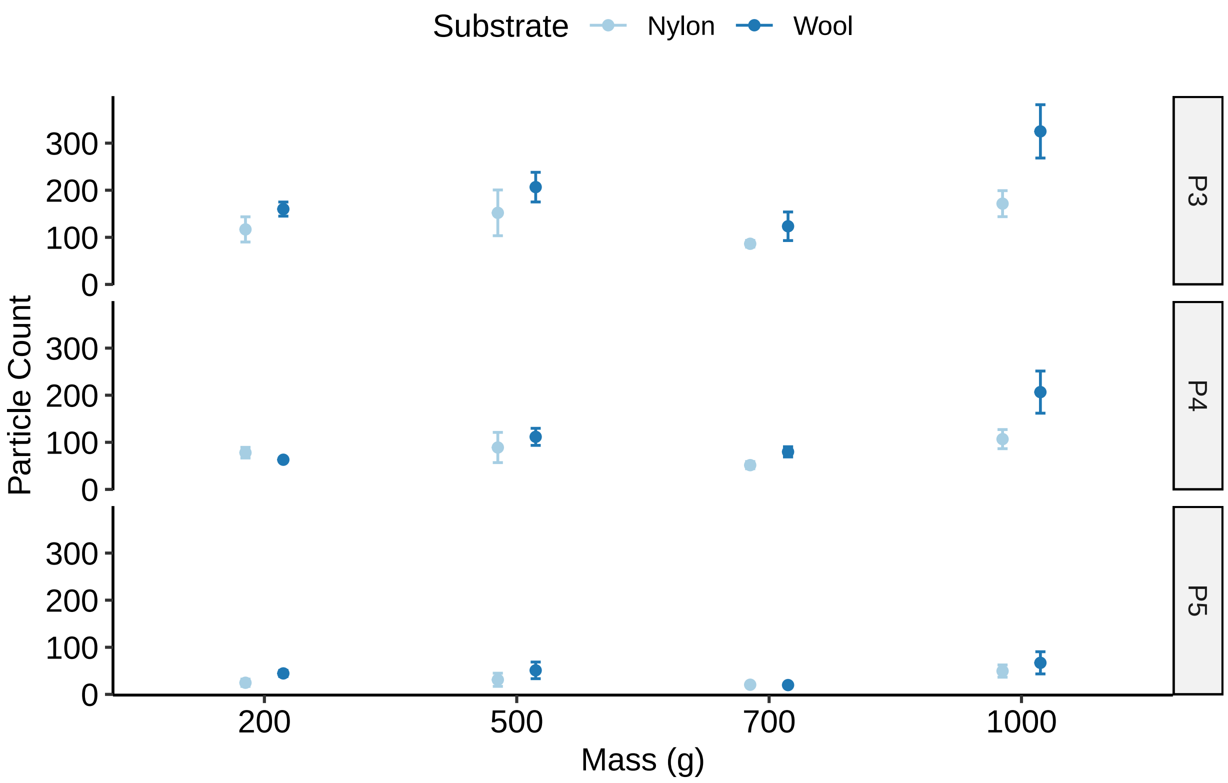


```r
m1000ny.dat = m1000.dat %>% filter(Substrate == "Nylon", Photo %in% c('P3', 'P4', 'P5'))

ggplot(m1000ny.dat, aes(x = numericFactorOrder(TransferTime), y = mean, colour = Note)) +
  labs(title = "Comparison of Camera Settings with Nylon",
       subtitle = "Broken down by photo",
       x = "Transfer Time (s)",
       y = "Particle Count",
       colour = "Camera\nSettings",
       caption = "Error bars: Std err") +
  scale_colour_brewer(palette = 'Paired') +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.08, position = position_dodge(width
  facet_grid(Photo ~ .) +
  facet_theme +
  theme_pubr()
```



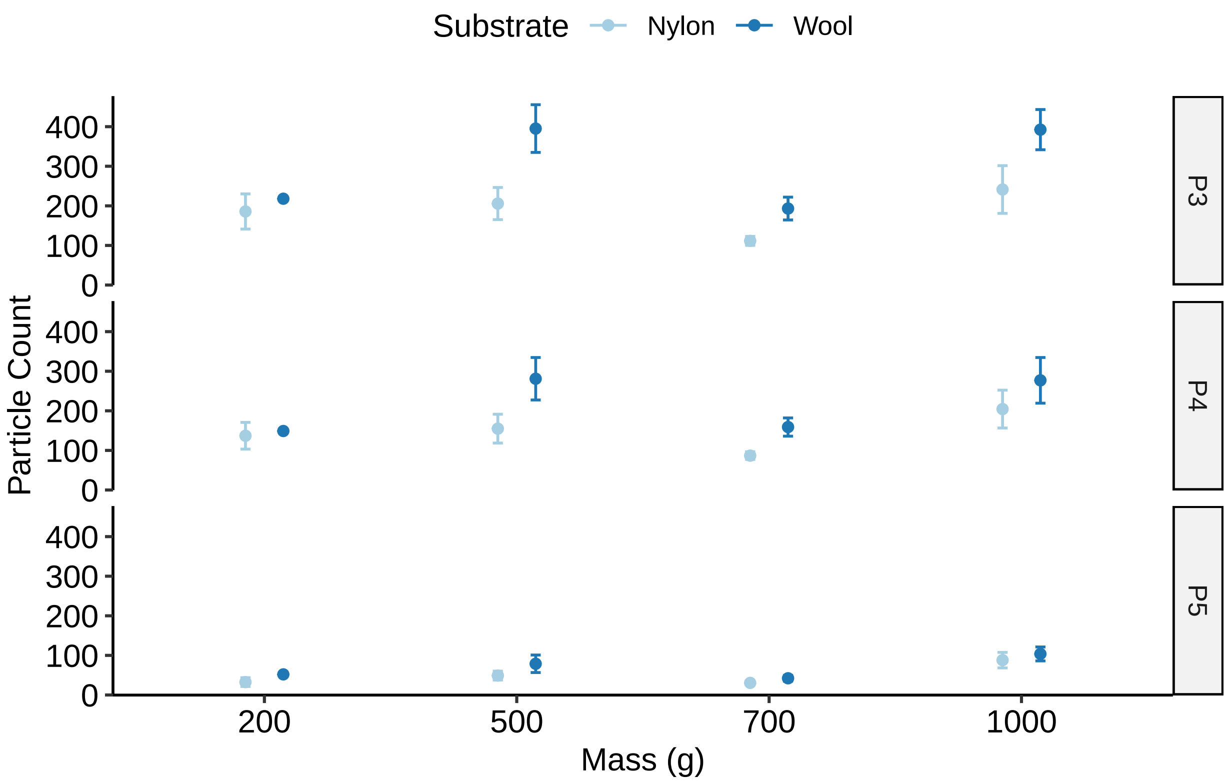
## Compare Camera Setting by Substrates

```
# compare camera settings by substrate type

c1.dat = t60.dat %>% filter(Note == 'C1', Photo %in% c('P3', 'P4', 'P5'))

c1plt = ggplot(c1.dat, aes(x = numericFactorOrder(Mass), y = mean, colour = Substrate)) +
  labs(
       x = 'Mass (g)',
       y = "Particle Count",
       colour = "Substrate",
       caption = "Error bars: Std err") +
  scale_color_brewer(palette = 'Paired') +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.08, position = position_dodge(width
  facet_grid(Photo ~ .) +
  facet_theme +
  theme_pubr()

c1plt
```



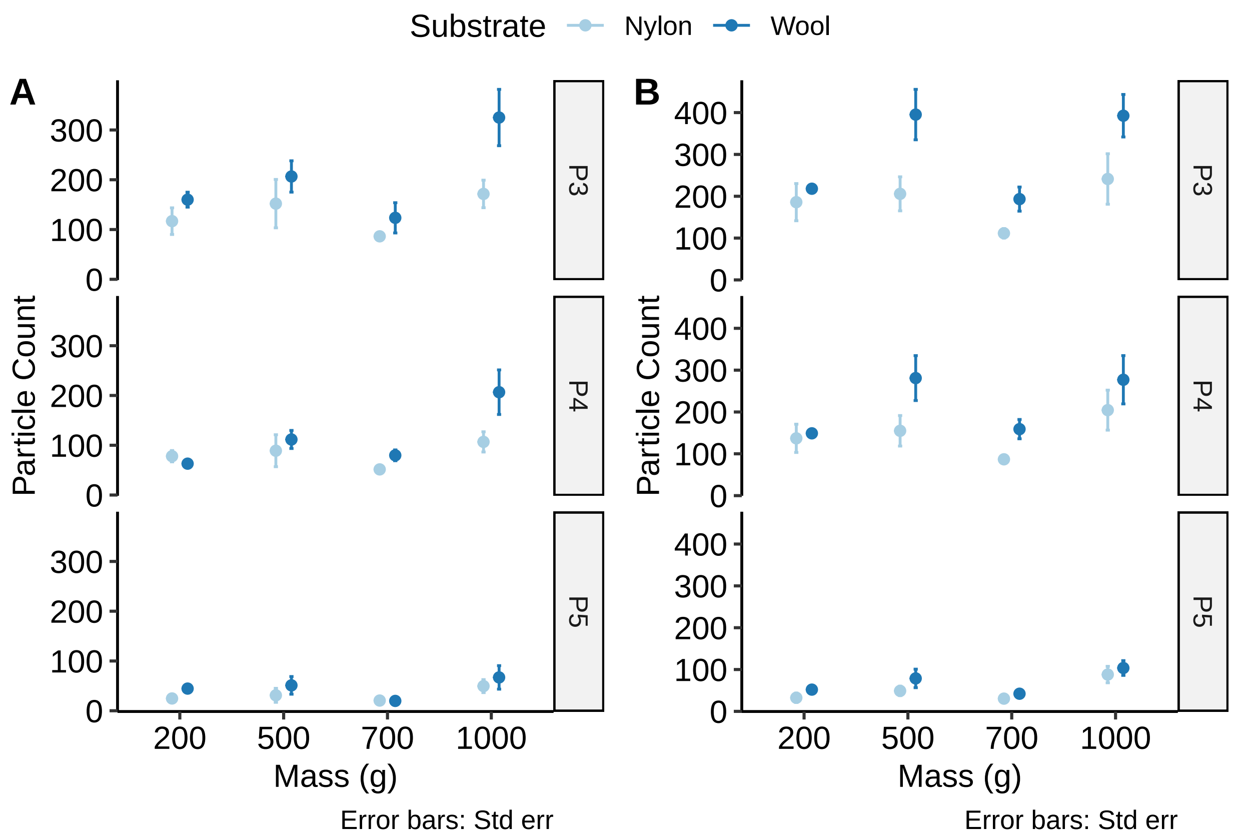
Error bars: Std err

```
# compare camera settings by substrate type

c2.dat = t60.dat %>% filter(Note == 'C2', Photo %in% c('P3', 'P4', 'P5'))

c2plt = ggplot(c2.dat, aes(x = numericFactorOrder(Mass), y = mean, colour = Substrate)) +
  labs(
       x = 'Mass (g)',
       y = "Particle Count",
       colour = "Substrate",
       caption = "Error bars: Std err") +
  scale_color_brewer(palette = 'Paired') +
  geom_point(position = position_dodge(width = 0.3)) +
  geom_errorbar(aes(ymin = mean - se, ymax = mean + se), width = 0.08, position = position_dodge(width =
  facet_grid(Photo ~ .) +
  facet_theme +
  theme_pubr()

c2plt
```



Error bars: Std err

```
c12subsplt = ggarrange(c1plt, c2plt, labels = c('A', 'B'), ncol = 2,
          common.legend = TRUE, legend = 'top')

c12subsplt
```



```
#ggsave('extcamera2.png', c12subsplt, height = 4.5, units = 'in')
```

Do Camera Settings Make a Statistical Difference?

This is count data which won't be normally distributed, so we'll go with the non-parametric Mann-Whitney test.

What we want to test is the effect of camera settings C1 & C2 on the particle counts on the substrates. We know that the counts are dependent on the substrate and photo number at the very least and possible mass and transfer time.

```
# get all data for photos 3, 4 & 5 exluding time 10
p345.dat = tlong.dat %>%
  filter(TransferTime != 10, Photo %in% c('P3', 'P4', 'P5')) %>%
  select(Substrate, Photo, Mass, TransferTime, Note, Count)
# for all the combinations of substrate, mass, transfer time and photo
# calculate the wilcoxon p-value
wt.dat = p345.dat %>%
  group_by(Substrate, Mass, TransferTime, Photo) %>%
  do(wt = wilcox.test(Count ~ Note, data = .)) %>%
  tidy(wt) %>%
  select(Substrate, Photo, Mass, TransferTime, statistic, p.value)

# as we done a lot of pairwise tests do a multiple hypothesis correction.
wt.dat = cbind(wt.dat, p.adj = p.adjust(wt.dat$p.value, method = 'BH'))
knitr::kable(head(wt.dat[order(wt.dat$p.value),], 10), caption = "Most significantly affected counts")
```

Table 5: Most significantly affected counts

Substrate	Photo	Mass	TransferTime	statistic	p.value	p.adj
Wool	P4	1000	120	0.0	0.0079365	0.3463203
Nylon	P4	700	60	4.0	0.0259740	0.3463203
Wool	P3	1000	120	2.0	0.0317460	0.3463203
Wool	P5	700	60	1.0	0.0317460	0.3463203
Wool	P4	700	60	1.5	0.0490901	0.3463203
Nylon	P3	700	60	5.5	0.0542411	0.3463203
Wool	P3	500	60	3.0	0.0555556	0.3463203
Nylon	P5	700	60	6.0	0.0636694	0.3463203
Wool	P3	1000	240	6.0	0.0649351	0.3463203
Wool	P5	1000	240	6.5	0.0776483	0.3727119

What if we take out the transfer time and mass factors? From the transfer analysis we saw that there isn't really any relationship between time and mass vs count.

```
# get all data for photos 3, 4 & 5 exluding time 10
p345.dat = tlong.dat %>%
  filter(TransferTime != 10, Substrate != 'Elas01', Photo %in% c('P3', 'P4', 'P5')) %>%
  select(Substrate, Photo, Note, Count)
# for all the combinations of substrate and photo
# calculate the wilcoxon p-value
wt.dat = p345.dat %>%
  group_by(Substrate, Photo) %>%
  do(wt = wilcox.test(Count ~ Note, data = .)) %>%
  tidy(wt) %>%
  select(Substrate, Photo, statistic, p.value)

# as we done a lot of pairwise tests do a multiple hypothesis correction.
wt.dat = cbind(wt.dat, p.adj = p.adjust(wt.dat$p.value, method = 'BH'))
knitr::kable(wt.dat, caption = "Most significantly affected counts")
```

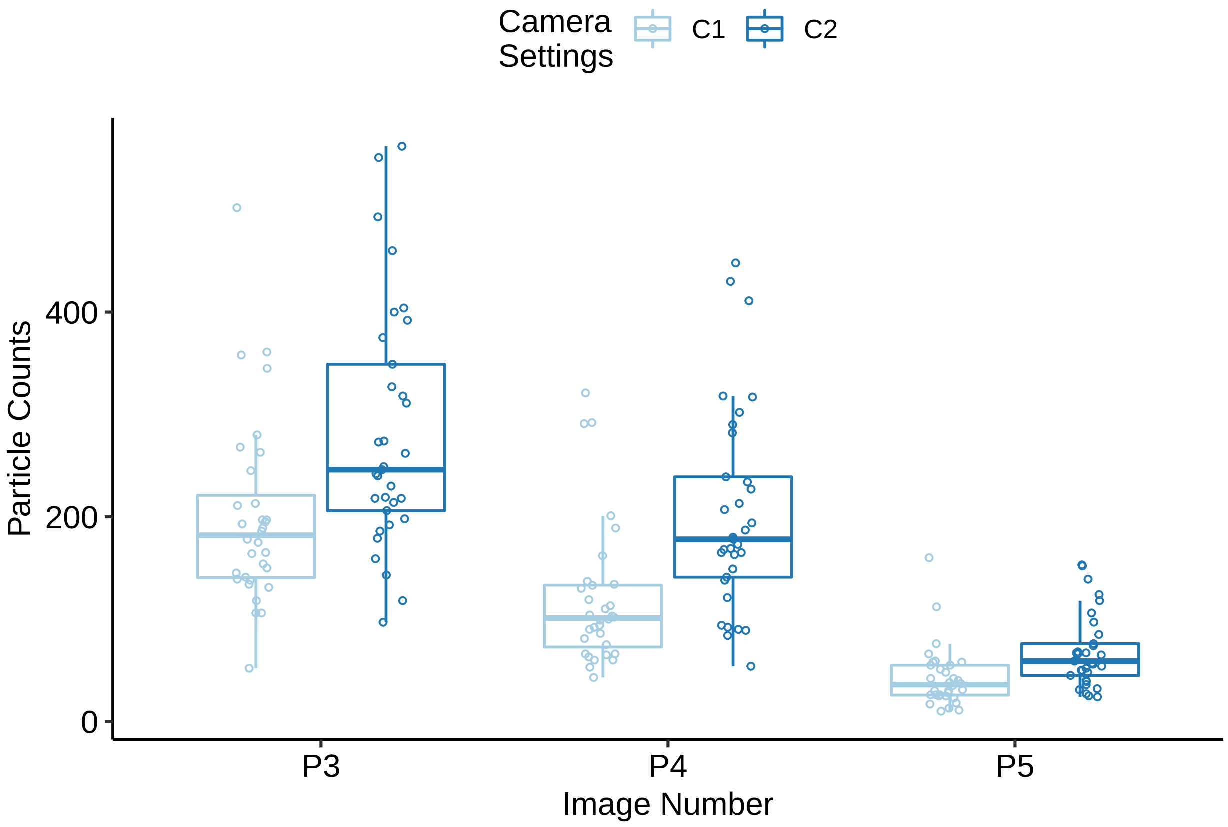
Table 6: Most significantly affected counts

Substrate	Photo	statistic	p.value	p.adj
Nylon	P3	516.0	0.0990024	0.1188029
Nylon	P4	432.0	0.0099738	0.0149607
Nylon	P5	525.5	0.1222979	0.1222979
Wool	P3	276.0	0.0009659	0.0019319
Wool	P4	232.5	0.0001083	0.0006500
Wool	P5	265.0	0.0005707	0.0017121

Surprise! All the wool data are now significant even after correction. Let's see what that looks like.

```
wsig.dat = p345.dat %>% filter(Substrate == 'Wool')

sigplt = ggplot(wsig.dat, aes(x = Photo, y = Count, colour = Note)) +
  labs(
       x = "Image Number",
       y = "Particle Counts",
      colour = 'Camera\nSettings') +
  scale_color_brewer(palette = 'Paired') +
  geom_boxplot(outlier.shape = NA) +
  geom_jitter(aes(colour = Note),position = position_jitterdodge(jitter.width = 0.1, dodge.width = 0.8)
  mytheme +
  theme_pubr()
```

```
sigplt
```

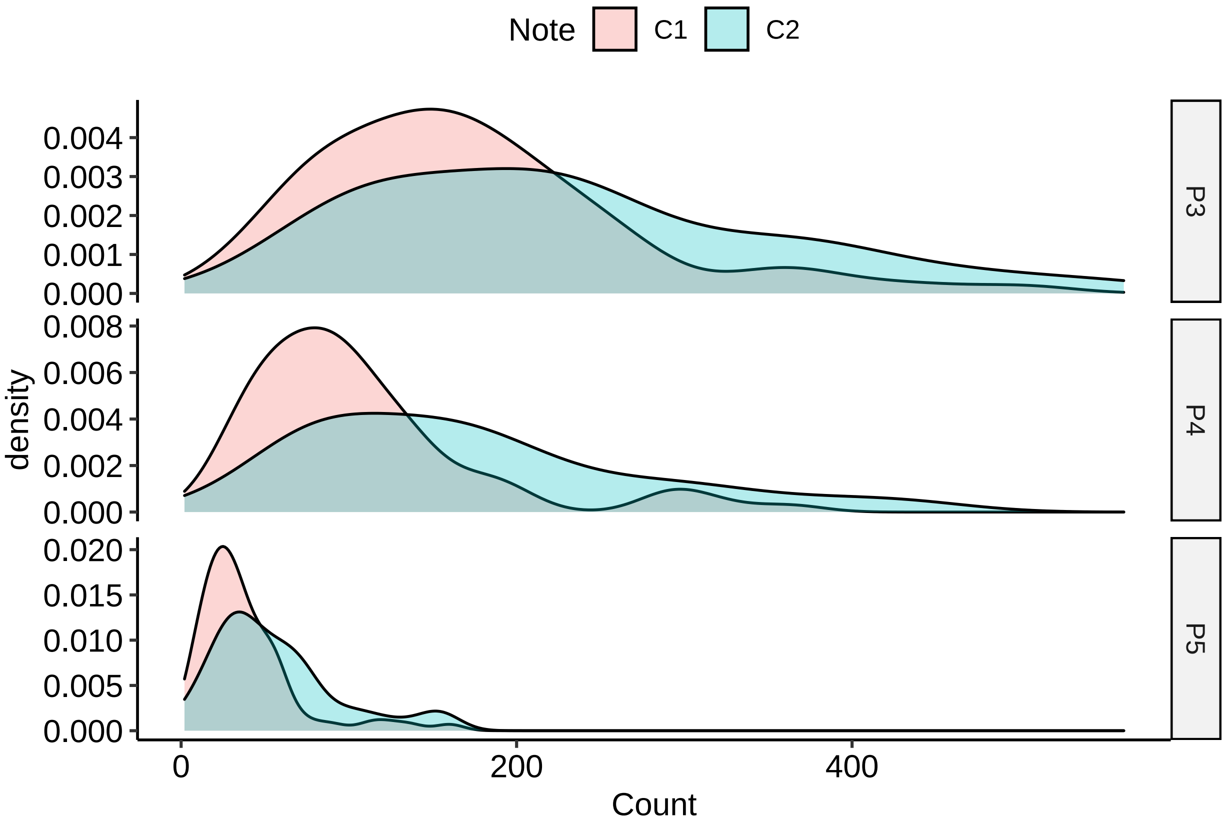


```
ggsave('expcamera3.png', sigplt, height = 4, width = 5.5, units = 'in')
```

So, does this mean that camera settings may need to be substrate-specific? Maybe. More data required. Natch!

Attempt at looking at distributions.

```
ggplot(p345.dat, aes(x = Count, fill = Note)) +
  geom_density(alpha = 0.3) +
  facet_grid(Photo ~ ., scales = 'free_y') +
  theme_pubr()
```


Note
C1
C2
density
0.004
0.003
0.002
0.001
0.000
0.008
0.006
0.004
0.002
0.000
0.020
0.015
0.010
0.005
0.000
P3
P4
P5
0
200
400
Count