Final Project

Ben Chaloupka, Tess Sameshima, and Scott Wallner

10/26/2020

Import Data

```
fractal_data <- clean_names(import(here("data", "frac_total.csv")))

raw_A <- clean_names(import(here("data", "frac_SA.csv")))

raw_B <- clean_names(import(here("data", "frac_SB.csv")))
```

Clean Data

```
fractal_data <- fractal_data %>%
  pivot_longer(cols = a:p,
               names_to = "participant",
               values_to = "dwell_time")

fractal_data <- fractal_data %>%
  mutate(direction = factor(direction, labels = c("Decay", "Growth", "Random")),
         participant = factor(participant),
         disp_image = factor(disp_image))

fractal_data <- fractal_data %>%
  mutate(fractal_type = case_when(
    grepl("Fern_A", disp_image, fixed = T) ~ "Fern_A",
    grepl("Fern_B", disp_image, fixed = T) ~ "Fern_B",
    grepl("Fractal_Curves_A", disp_image, fixed = T) ~ "Fractal_Curves_A",
    grepl("Fractal_Curves_B", disp_image, fixed = T) ~ "Fractal_Curves_B",
    grepl("Fractal_Curves_C", disp_image, fixed = T) ~ "Fractal_Curves_C",
    grepl("Fractal_Curves_D", disp_image, fixed = T) ~ "Fractal_Curves_D",
    grepl("Fractal_Curves_E", disp_image, fixed = T) ~ "Fractal_Curves_E",
    grepl("Hilbert", disp_image, fixed = T) ~ "Hilbert",
    grepl("Moore", disp_image, fixed = T) ~ "Moore",
    grepl("Pythagoras_A", disp_image, fixed = T) ~ "Pythagoras_A",
    grepl("Pythagoras_B", disp_image, fixed = T) ~ "Pythagoras_B",
    grepl("Sierpinski_A", disp_image, fixed = T) ~ "Sierpinski_A",
    grepl("Sierpinski_B", disp_image, fixed = T) ~ "Sierpinski_B",
    grepl("Tree_A", disp_image, fixed = T) ~ "Tree_A"
  ))

fractal_data <- fractal_data %>%
  mutate(sequence_position = case_when(
    grepl("A0.png", disp_image, fixed = T) ~ 1,
```

```
    grepl("B0.png", disp_image, fixed = T) ~ 1,
    grepl("C0.png", disp_image, fixed = T) ~ 1,
    grepl("D0.png", disp_image, fixed = T) ~ 1,
    grepl("E0.png", disp_image, fixed = T) ~ 1,
    grepl("_0.png", disp_image, fixed = T) ~ 1,
    grepl("A100.png", disp_image, fixed = T) ~ 2,
    grepl("B100.png", disp_image, fixed = T) ~ 2,
    grepl("A1.png", disp_image, fixed = T) ~ 2,
    grepl("B1.png", disp_image, fixed = T) ~ 2,
    grepl("C1.png", disp_image, fixed = T) ~ 2,
    grepl("D1.png", disp_image, fixed = T) ~ 2,
    grepl("E1.png", disp_image, fixed = T) ~ 2,
    grepl("_1.png", disp_image, fixed = T) ~ 2,
    grepl("A200.png", disp_image, fixed = T) ~ 3,
    grepl("B200.png", disp_image, fixed = T) ~ 3,
    grepl("A2.png", disp_image, fixed = T) ~ 3,
    grepl("B2.png", disp_image, fixed = T) ~ 3,
    grepl("C2.png", disp_image, fixed = T) ~ 3,
    grepl("D2.png", disp_image, fixed = T) ~ 3,
    grepl("E2.png", disp_image, fixed = T) ~ 3,
    grepl("_2.png", disp_image, fixed = T) ~ 3,
    grepl("A400.png", disp_image, fixed = T) ~ 4,
    grepl("B400.png", disp_image, fixed = T) ~ 4,
    grepl("A3.png", disp_image, fixed = T) ~ 4,
    grepl("B3.png", disp_image, fixed = T) ~ 4,
    grepl("C3.png", disp_image, fixed = T) ~ 4,
    grepl("D3.png", disp_image, fixed = T) ~ 4,
    grepl("E3.png", disp_image, fixed = T) ~ 4,
    grepl("_3.png", disp_image, fixed = T) ~ 4,
    grepl("A800.png", disp_image, fixed = T) ~ 5,
    grepl("B800.png", disp_image, fixed = T) ~ 5,
    grepl("A4.png", disp_image, fixed = T) ~ 5,
    grepl("B4.png", disp_image, fixed = T) ~ 5,
    grepl("C4.png", disp_image, fixed = T) ~ 5,
    grepl("D4.png", disp_image, fixed = T) ~ 5,
    grepl("E4.png", disp_image, fixed = T) ~ 5,
    grepl("_4.png", disp_image, fixed = T) ~ 5,
    grepl("A1600.png", disp_image, fixed = T) ~ 6,
    grepl("B1600.png", disp_image, fixed = T) ~ 6,
    grepl("A5.png", disp_image, fixed = T) ~ 6,
    grepl("B5.png", disp_image, fixed = T) ~ 6,
    grepl("C5.png", disp_image, fixed = T) ~ 6,
    grepl("D5.png", disp_image, fixed = T) ~ 6,
    grepl("E5.png", disp_image, fixed = T) ~ 6,
    grepl("_5.png", disp_image, fixed = T) ~ 6,
    grepl("A3200.png", disp_image, fixed = T) ~ 7,
    grepl("B3200.png", disp_image, fixed = T) ~ 7,
    grepl("A6.png", disp_image, fixed = T) ~ 7,
    grepl("A6400.png", disp_image, fixed = T) ~ 8,
    grepl("B6400.png", disp_image, fixed = T) ~ 8,
    grepl("A7.png", disp_image, fixed = T) ~ 8,
    grepl("A12800.png", disp_image, fixed = T) ~ 9,
    grepl("B12800.png", disp_image, fixed = T) ~ 9
```

```
  ))

raw_A_long <- raw_A %>%
 pivot_longer(cols = a:f, names_to = "participant", values_to = "dwell_time")

raw_B_long <- raw_B %>%
 pivot_longer(cols = g:p, names_to = "participant", values_to = "dwell_time")
```

Create New Dataset Excluding Outliers

```
cutoff <- mean(fractal_data$dwell_time) + 3 * sd(fractal_data$dwell_time)

filtered_fractal_data <- fractal_data %>%
  filter(dwell_time < cutoff)
```

Create Dataset with Difference Between Growth and Decay Dwell Times as its own Variable

```
fractal_data_wide <- fractal_data %>%
  pivot_wider(names_from = direction,
              values_from = dwell_time) %>%
  select(-Random) %>%
  mutate(Growth = unlist(Growth),
         Decay = unlist(Decay),
         diff_dwell_time = Growth - Decay)
```

```
## Warning: Values are not uniquely identified; output will contain list-cols.
## * Use `values_fn = list` to suppress this warning.
## * Use `values_fn = length` to identify where the duplicates arise
## * Use `values_fn = {summary_fun}` to summarise duplicates
```

```
fractal_data_wide %>%
  group_by(participant) %>%
  summarise(mean_diff_dwell_time = mean(diff_dwell_time)) %>%
  knitr::kable(col.names = c("Participant", "Mean Difference in Dwell Times between Growth and Decay"))
```

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

Participant	Mean Difference in Dwell Times between Growth and Decay
a	2.0253251
b	0.4027316
c	0.0727060
d	0.0262463
e	0.0260263
f	0.0468790
g	0.0172402
h	0.0201722
i	0.0148214
j	0.0643832
k	0.0784663
l	0.1487259
m	0.2542457
n	0.0746982

Participant	Mean Difference in Dwell Times between Growth and Decay
o	0.0113360
p	0.0746982

Descriptive Stats by Participant

```
fractal_data %>%
  group_by(participant) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
            sd_dwell_time = round(sd(dwell_time), 3)) %>%
  knitr::kable(col.names = c("Participant", "Dwell Time Mean", "Dwell Time SD"))
```

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

Participant	Dwell Time Mean	Dwell Time SD
a	1.955	1.599
b	1.053	0.678
c	0.720	0.237
d	0.481	0.161
e	0.796	0.117
f	0.759	0.132
g	0.667	0.149
h	0.543	0.153
i	1.169	0.357
j	1.087	0.362
k	1.075	0.431
l	0.697	0.224
m	0.766	0.286
n	0.624	0.181
o	0.419	0.053
p	0.770	0.292

Descriptive Statistics by Direction

```
fractal_data %>%
  group_by(direction) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
            sd_dwell_time = round(sd(dwell_time), 3)) %>%
  knitr::kable(col.names = c("Direction", "Dwell Time Mean", "Dwell Time SD"))
```

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

Direction	Dwell Time Mean	Dwell Time SD
Decay	0.736	0.457
Growth	0.946	0.902
Random	0.857	0.464

Descriptive Statistics by Fractal

```
fractal_data %>%
  group_by(disp_image) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
            sd_dwell_time = round(sd(dwell_time), 3)) %>%
  arrange(mean_dwell_time) %>%
  knitr::kable(col.names = c("Fractal", "Dwell Time Mean", "Dwell Time SD"))
```

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

Fractal	Dwell Time Mean	Dwell Time SD
Images/Fern_B800.png	0.730	0.326
Images/Moore_0.png	0.738	0.241
Images/Sierpinski_B1.png	0.740	0.266
Images/Hilbert_0.png	0.745	0.259
Images/Sierpinski_A0.png	0.746	0.260
Images/Sierpinski_A2.png	0.756	0.276
Images/Sierpinski_A1.png	0.757	0.311
Images/Sierpinski_A3.png	0.757	0.310
Images/Fern_B200.png	0.761	0.409
Images/Fern_A0.png	0.762	0.272
Images/Fractal_Curves_D0.png	0.763	0.285
Images/Fern_B1600.png	0.765	0.402
Images/Fern_B400.png	0.770	0.360
Images/Sierpinski_B4.png	0.773	0.420
Images/Fern_A800.png	0.781	0.758
Images/Fern_A1600.png	0.782	0.653
Images/Fractal_Curves_B0.png	0.782	0.318
Images/Sierpinski_B2.png	0.782	0.333
Images/Fern_A3200.png	0.785	0.564
Images/Moore_2.png	0.785	0.473
Images/Sierpinski_B0.png	0.786	0.293
Images/Fern_A400.png	0.787	0.685
Images/Fern_A200.png	0.788	0.786
Images/Moore_1.png	0.796	0.385
Images/Fractal_Curves_E2.png	0.797	0.564
Images/Fern_B0.png	0.799	0.265
Images/Sierpinski_B3.png	0.799	0.367
Images/Hilbert_1.png	0.800	0.434
Images/Fractal_Curves_D1.png	0.803	0.413
Images/Fractal_Curves_E5.png	0.804	0.381
Images/Pythagoras_B0.png	0.804	0.366
Images/Fern_B3200.png	0.806	0.460
Images/Moore_4.png	0.806	0.495
Images/Hilbert_2.png	0.807	0.615
Images/Fractal_Curves_E0.png	0.811	0.297
Images/Fractal_Curves_C0.png	0.813	0.385
Images/Fern_A100.png	0.814	0.593
Images/Fractal_Curves_E1.png	0.814	0.489
Images/Fractal_Curves_D2.png	0.815	0.465
Images/Fractal_Curves_D3.png	0.815	0.527
Images/Fractal_Curves_C1.png	0.816	0.489
Images/Pythagoras_B1.png	0.824	0.438

Fractal	Dwell Time Mean	Dwell Time SD
Images/Pythagoras_A0.png	0.825	0.432
Images/Fractal_Curves_A2.png	0.827	0.558
Images/Fern_B100.png	0.828	0.604
Images/Tree_A3.png	0.828	0.632
Images/Hilbert_3.png	0.832	0.522
Images/Tree_A0.png	0.832	0.397
Images/Fractal_Curves_B1.png	0.835	0.610
Images/Fractal_Curves_E4.png	0.835	0.636
Images/Tree_A2.png	0.835	0.443
Images/Tree_A4.png	0.839	0.583
Images/Fractal_Curves_A0.png	0.842	0.445
Images/Tree_A1.png	0.859	0.436
Images/Fractal_Curves_D4.png	0.861	0.598
Images/Moore_5.png	0.863	0.509
Images/Fractal_Curves_A1.png	0.867	0.663
Images/Pythagoras_A1.png	0.874	0.811
Images/Sierpinski_A4.png	0.876	0.458
Images/Moore_3.png	0.877	0.730
Images/Tree_A5.png	0.884	0.586
Images/Fern_B6400.png	0.889	0.860
Images/Sierpinski_B5.png	0.893	0.537
Images/Hilbert_5.png	0.894	0.635
Images/Fern_B12800.png	0.897	0.468
Images/Fractal_Curves_B2.png	0.904	0.786
Images/Hilbert_4.png	0.913	0.737
Images/Fractal_Curves_E3.png	0.915	1.117
Images/Fractal_Curves_C4.png	0.922	0.737
Images/Fractal_Curves_C2.png	0.934	0.744
Images/Fern_A6400.png	0.940	0.946
Images/Fractal_Curves_D5.png	0.942	0.689
Images/Fractal_Curves_B5.png	0.947	0.607
Images/Tree_A7.png	0.953	0.631
Images/Pythagoras_B2.png	0.955	0.671
Images/Fractal_Curves_B4.png	0.960	1.014
Images/Fractal_Curves_A3.png	0.966	0.890
Images/Pythagoras_A2.png	0.966	0.718
Images/Pythagoras_A3.png	0.966	0.644
Images/Tree_A6.png	0.971	0.756
Images/Fractal_Curves_B3.png	0.975	1.047
Images/Fractal_Curves_A5.png	0.997	0.915
Images/Fractal_Curves_C5.png	0.998	0.925
Images/Fern_A12800.png	1.005	1.048
Images/Fractal_Curves_A4.png	1.006	1.099
Images/Fractal_Curves_C3.png	1.008	0.845
Images/Pythagoras_B3.png	1.054	0.850

Descriptive Statistics by Participant and Direction

```
fractal_data %>%
  group_by(participant, direction) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
```

```
            sd_dwell_time = round(sd(dwell_time), 3)) %>%
  knitr::kable(col.names = c("Participant", "Direction", "Dwell Time Mean", "Dwell Time SD"))
```

```
## `summarise()` regrouping output by 'participant' (override with `.groups` argument)
```

Participant	Direction	Dwell Time Mean	Dwell Time SD
a	Decay	1.414	1.151
a	Growth	3.439	1.912
a	Random	1.482	1.083
b	Decay	1.156	0.503
b	Growth	1.558	0.965
b	Random	0.750	0.328
c	Decay	0.623	0.134
c	Growth	0.696	0.228
c	Random	0.781	0.263
d	Decay	0.424	0.125
d	Growth	0.451	0.182
d	Random	0.525	0.154
e	Decay	0.756	0.125
e	Growth	0.782	0.116
e	Random	0.822	0.107
f	Decay	0.746	0.145
f	Growth	0.793	0.141
f	Random	0.748	0.117
g	Decay	0.595	0.070
g	Growth	0.612	0.077
g	Random	0.731	0.176
h	Decay	0.454	0.108
h	Growth	0.474	0.122
h	Random	0.622	0.145
i	Decay	1.210	0.415
i	Growth	1.225	0.307
i	Random	1.121	0.344
j	Decay	0.811	0.175
j	Growth	0.875	0.198
j	Random	1.332	0.326
k	Decay	0.841	0.134
k	Growth	0.920	0.194
k	Random	1.270	0.517
l	Decay	0.591	0.169
l	Growth	0.740	0.228
l	Random	0.728	0.230
m	Decay	0.752	0.308
m	Growth	1.006	0.323
m	Random	0.653	0.152
n	Decay	0.494	0.114
n	Growth	0.568	0.146
n	Random	0.716	0.173
o	Decay	0.418	0.062
o	Growth	0.429	0.070
o	Random	0.414	0.036
p	Decay	0.494	0.114
p	Growth	0.568	0.146

Participant	Direction	Dwell Time Mean	Dwell Time SD
p	Random	1.010	0.194

A Couple of Plots

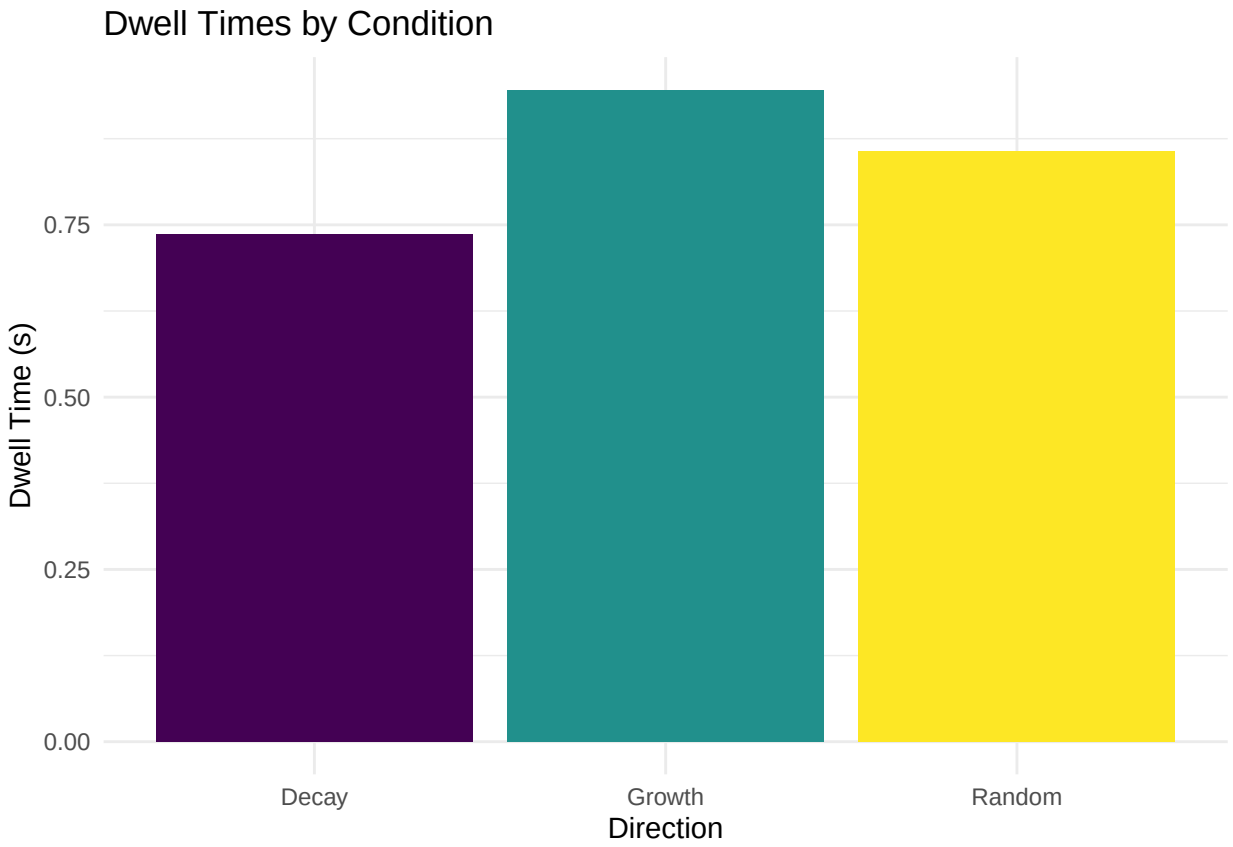
```
new_data_for_now <- fractal_data %>%
  group_by(participant, direction) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
            sd_dwell_time = round(sd(dwell_time), 3))
```

```
## `summarise()` regrouping output by 'participant' (override with `.groups` argument)
```

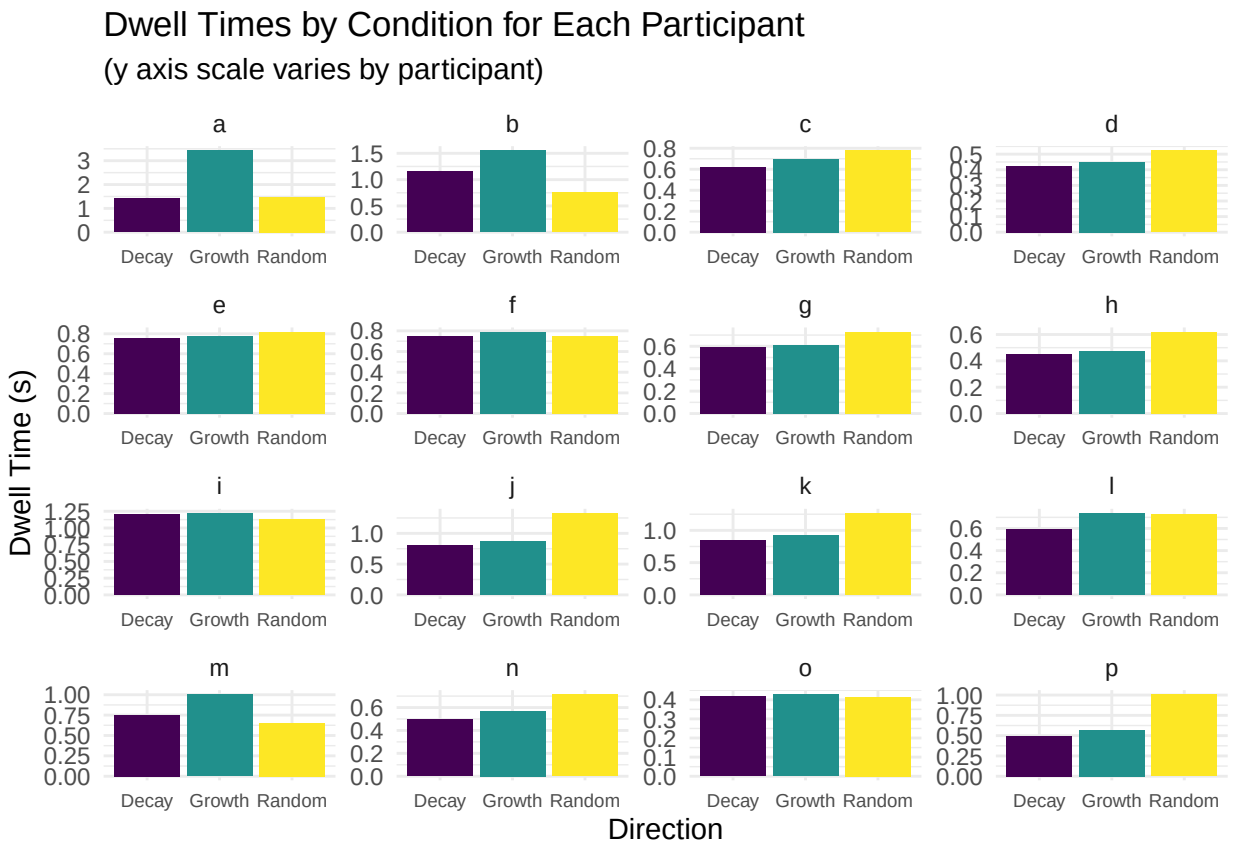
```
new_data_for_later <- fractal_data %>%
  group_by(direction) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
            sd_dwell_time = round(sd(dwell_time), 3))
```

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

```
fractal_data %>%
  ggplot(aes(x = direction, y = dwell_time, fill = direction)) +
  geom_col(data = new_data_for_later, aes(y = mean_dwell_time)) +
  labs(x = "Direction", y = "Dwell Time (s)", title = "Dwell Times by Condition") +
  theme(legend.position = "none") +
  scale_fill_viridis_d()
```



```
fractal_data %>%
  ggplot(aes(x = direction, y = dwell_time, fill = direction)) +
  geom_col(data = new_data_for_now, aes(y = mean_dwell_time)) +
  facet_wrap(~participant, scales = "free") +
  labs(x = "Direction", y = "Dwell Time (s)", title = "Dwell Times by Condition for Each Participant", s
  theme(legend.position = "none") +
  scale_fill_viridis_d() +
  theme(axis.text.x = element_text(size = 7))
```



Building a Model (and a Plot for that Model)

```
model1 <- lm(data = fractal_data, dwell_time ~ direction)

summary(model1)
```

```
##
## Call:
## lm(formula = dwell_time ~ direction, data = fractal_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7484 -0.2956 -0.1256  0.0914  7.5001
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.73613    0.01615  45.569  < 2e-16 ***
```

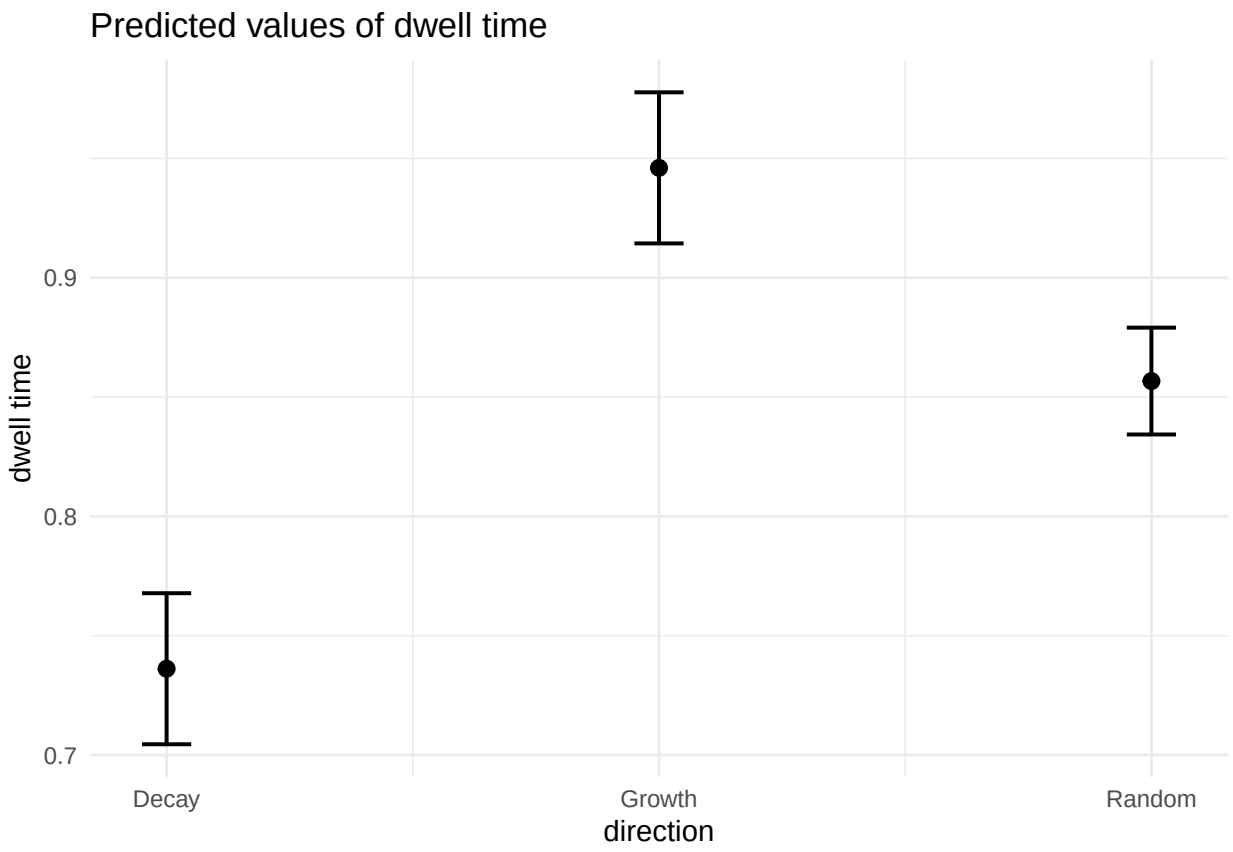
```
## directionGrowth  0.20992    0.02285   9.189  < 2e-16 ***
## directionRandom  0.12056    0.01978   6.094 1.18e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6027 on 5565 degrees of freedom
## Multiple R-squared:  0.01511,    Adjusted R-squared:  0.01475
## F-statistic: 42.68 on 2 and 5565 DF,  p-value: < 2.2e-16
```

```
car::Anova(model1)
```

```
## Anova Table (Type II tests)
##
## Response: dwell_time
##             Sum Sq   Df F value    Pr(>F)
## direction    31.01    2  42.681 < 2.2e-16 ***
## Residuals 2021.54 5565
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
sjPlot::plot_model(model1, type = "pred", terms = c("direction"))
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

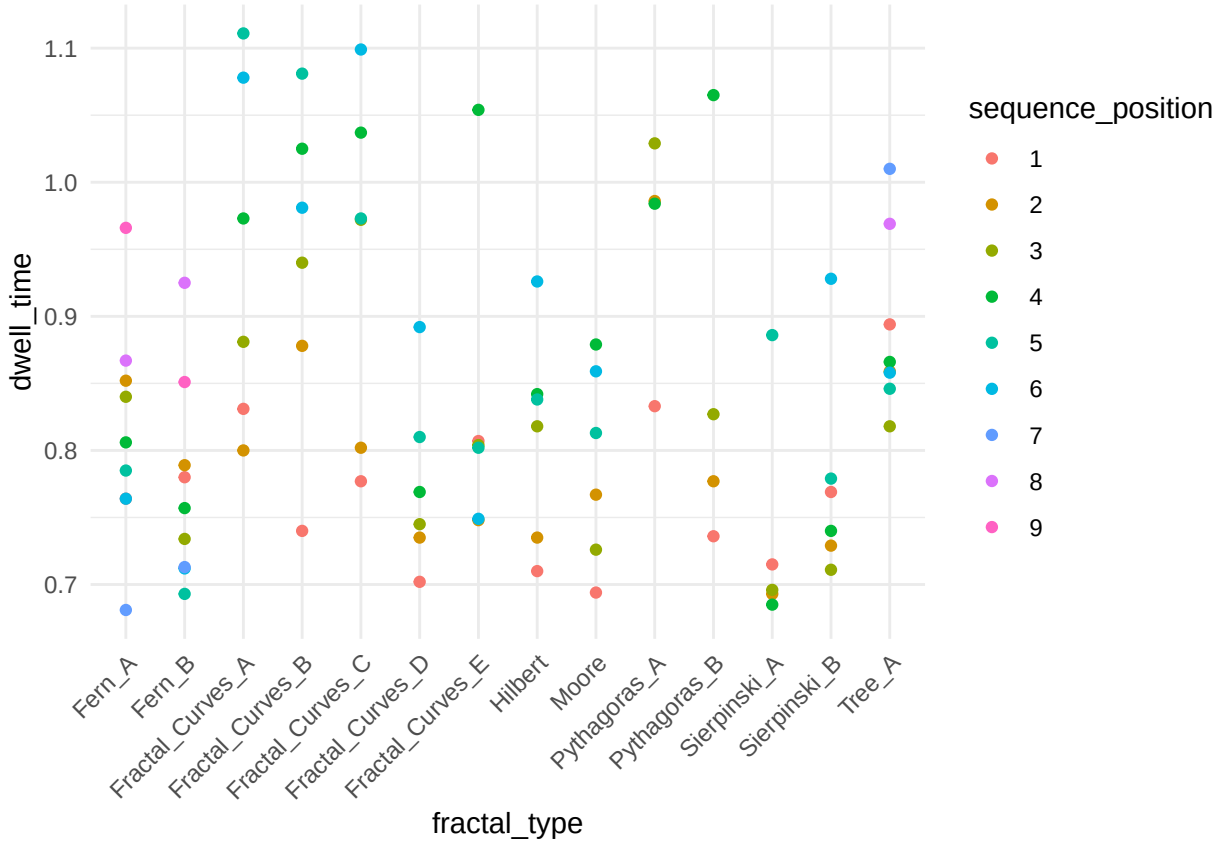


More Models and Plots

```
seq_fractal_data <- fractal_data %>%
  filter(direction != "Random") %>%
  mutate(sequence_position = factor(sequence_position),
         fractal_type = factor(fractal_type))

new_data_forever <- seq_fractal_data %>%
  group_by(sequence_position, fractal_type) %>%
  summarise(mean_dwell_time = round(mean(dwell_time), 3),
            sd_dwell_time = round(sd(dwell_time), 3))
```

```
## `summarise()` regrouping output by 'sequence_position' (override with `.groups` argument)
```

```
seq_fractal_data %>% # change points to numbers for easy sequence position identification
  ggplot(aes(x = fractal_type, y = dwell_time, color = sequence_position)) +
  geom_point(data = new_data_forever, aes(y = mean_dwell_time)) +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
```



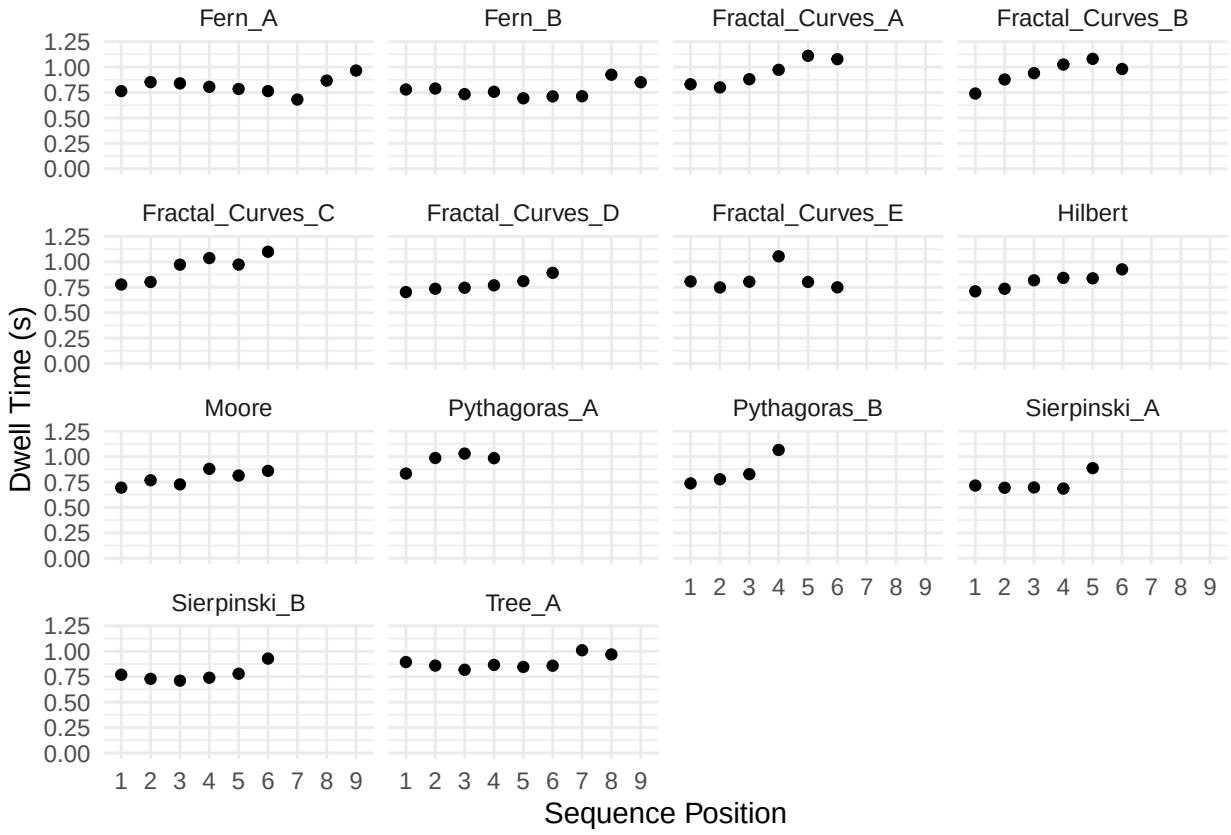

```
model2 <- lm(data = seq_fractal_data, dwell_time ~ sequence_position * fractal_type)

car::Anova(model2)
```

```
## Note: model has aliased coefficients
##       sums of squares computed by model comparison
```

```
## Anova Table (Type II tests)
##
## Response: dwell_time
```

```
##                                  Sum Sq   Df F value   Pr(>F)
## sequence_position                  8.48    8  2.0161 0.040992 *
## fractal_type                      15.70   13  2.2965 0.005115 **
## sequence_position:fractal_type    12.59   65  0.3684 0.999999
## Residuals                       1418.18 2697
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
seq_fractal_data %>%
  ggplot(aes(x = sequence_position, y = dwell_time)) +
  facet_wrap(~fractal_type) +
  geom_point(data = new_data_forever, aes(y = mean_dwell_time)) +
  scale_y_continuous(limits = c(0, 1.2)) +
  labs(y = "Dwell Time (s)", x = "Sequence Position")
```

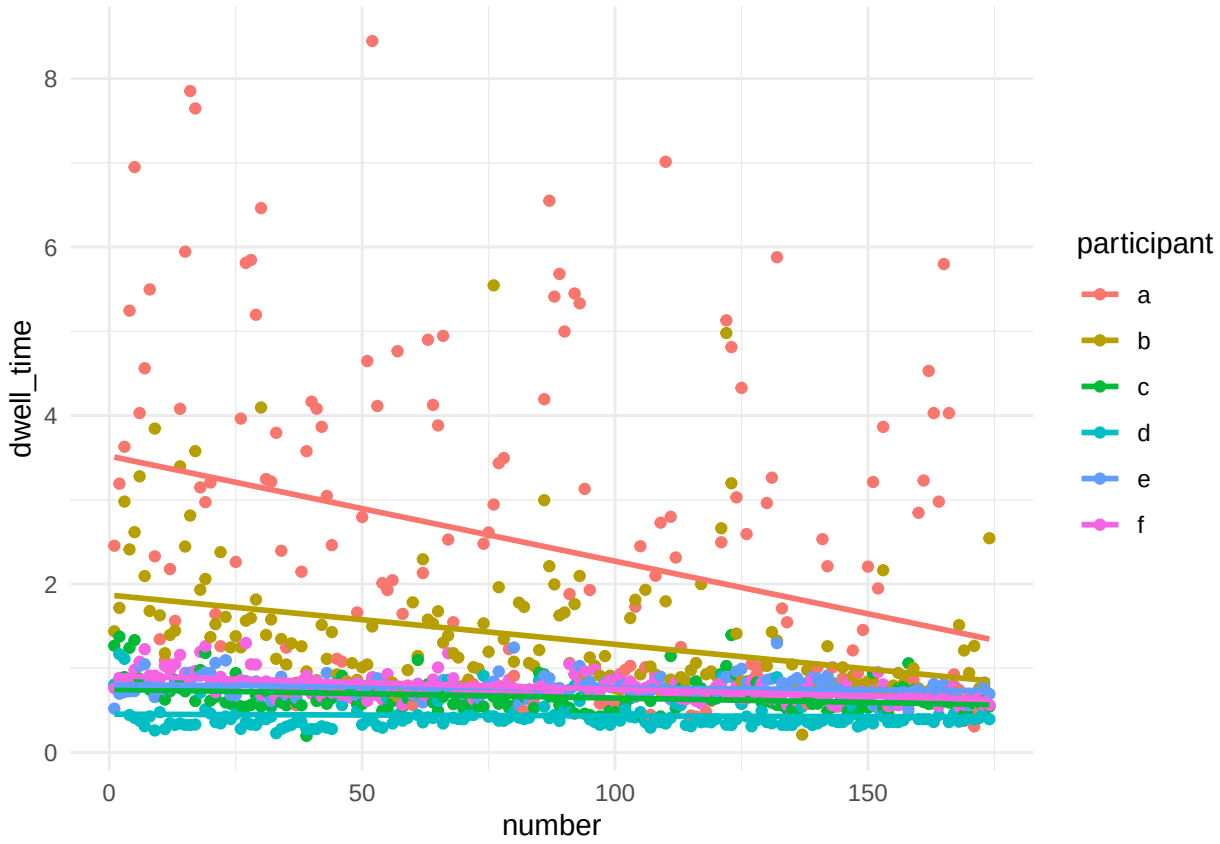


Dwell Time over Time (yeah, that's right)

These plots show how dwell time decreased over time as participants became bored, fatigued, etc. The order of the fractals was counterbalanced such that the A group and the B group saw the fractal types in the reverse order.

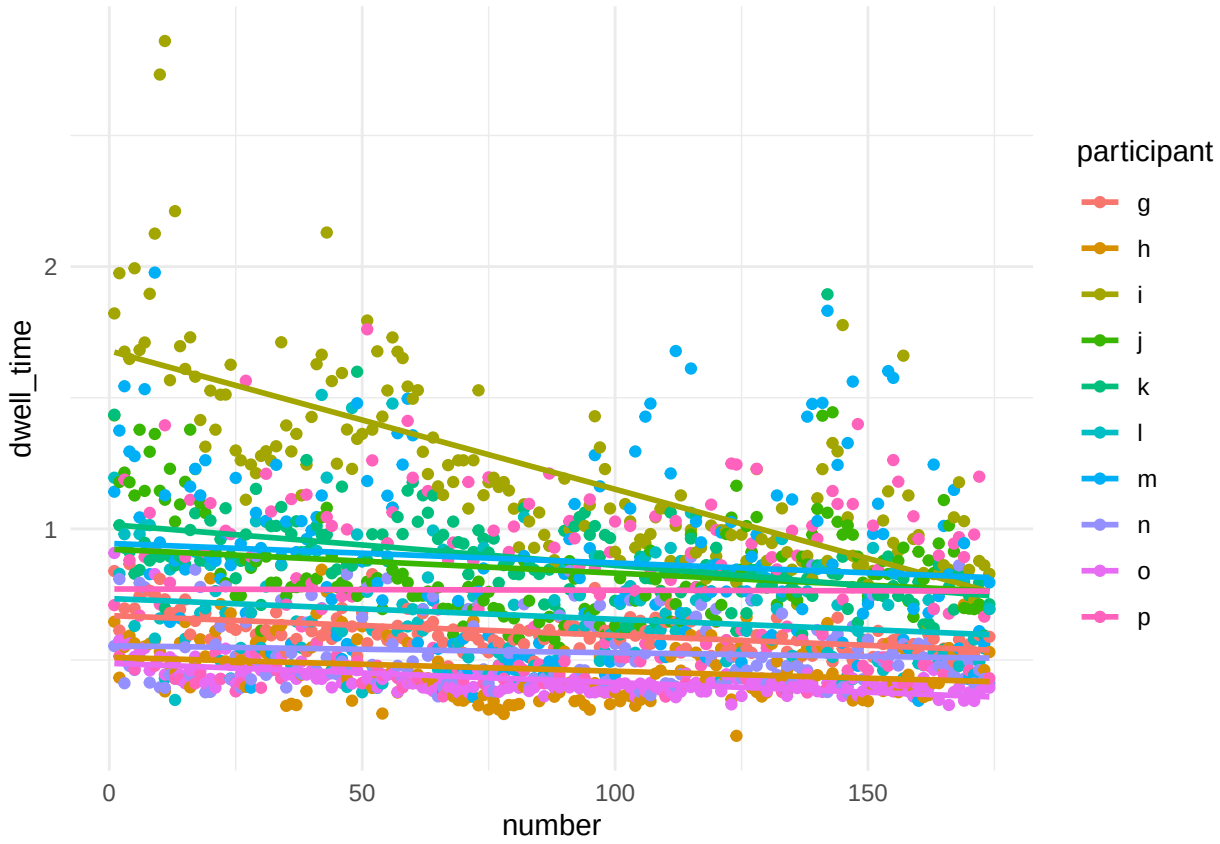
```
raw_A_long %>%
  ggplot(aes(x = number, y = dwell_time, color = participant)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



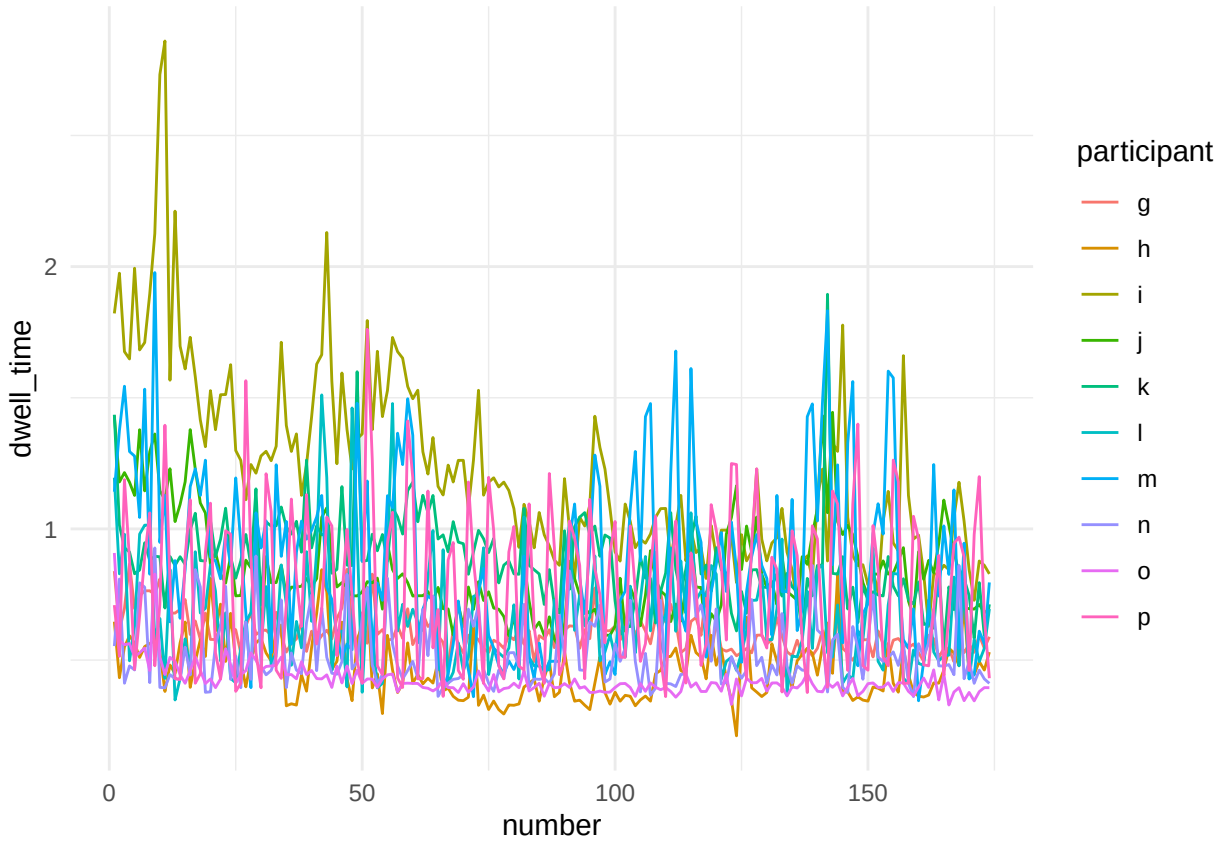

```
raw_B_long %>%
  ggplot(aes(x = number, y = dwell_time, color = participant)) +
  geom_point() +
  geom_smooth(method = "lm", se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



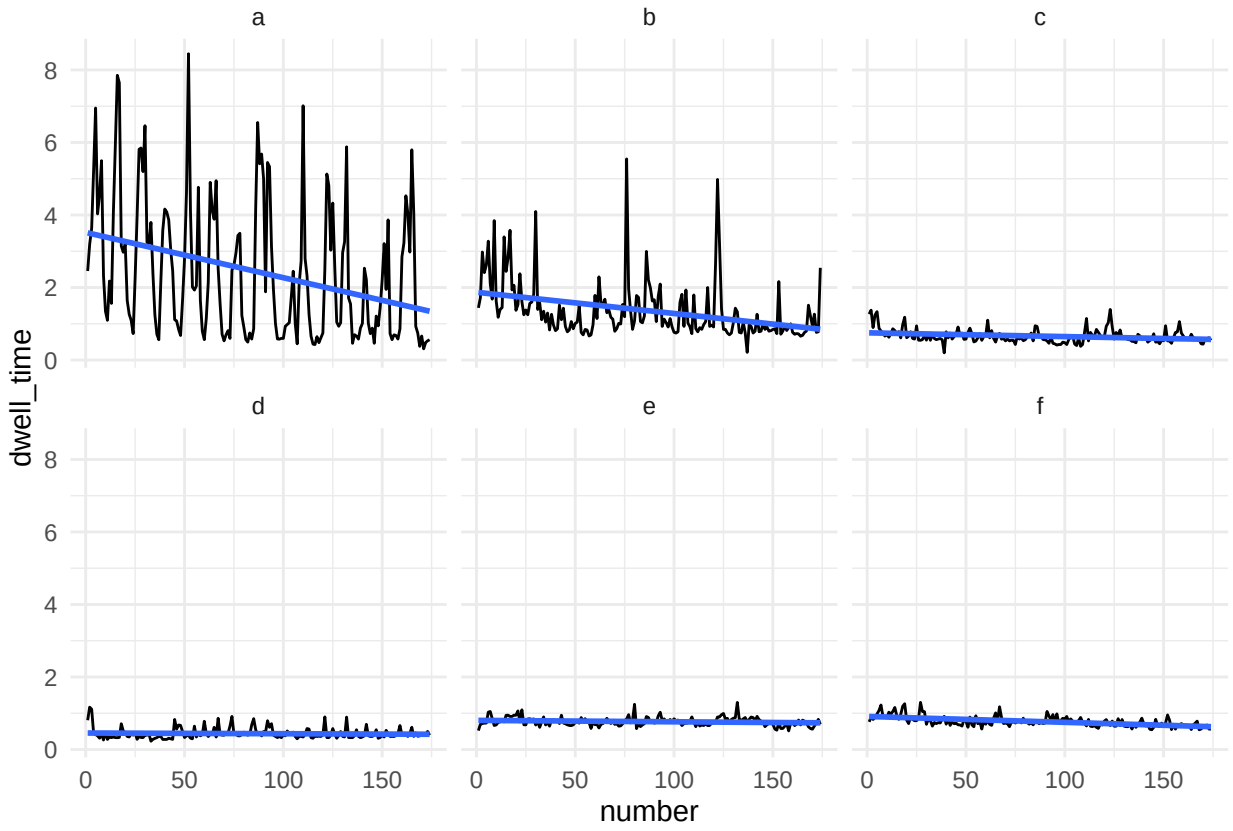

```
raw_A_long %>%
  ggplot(aes(x = number, y = dwell_time, color = participant)) +
  geom_line()
```


```
raw_B_long %>%
  ggplot(aes(x = number, y = dwell_time, color = participant)) +
  geom_line()
```



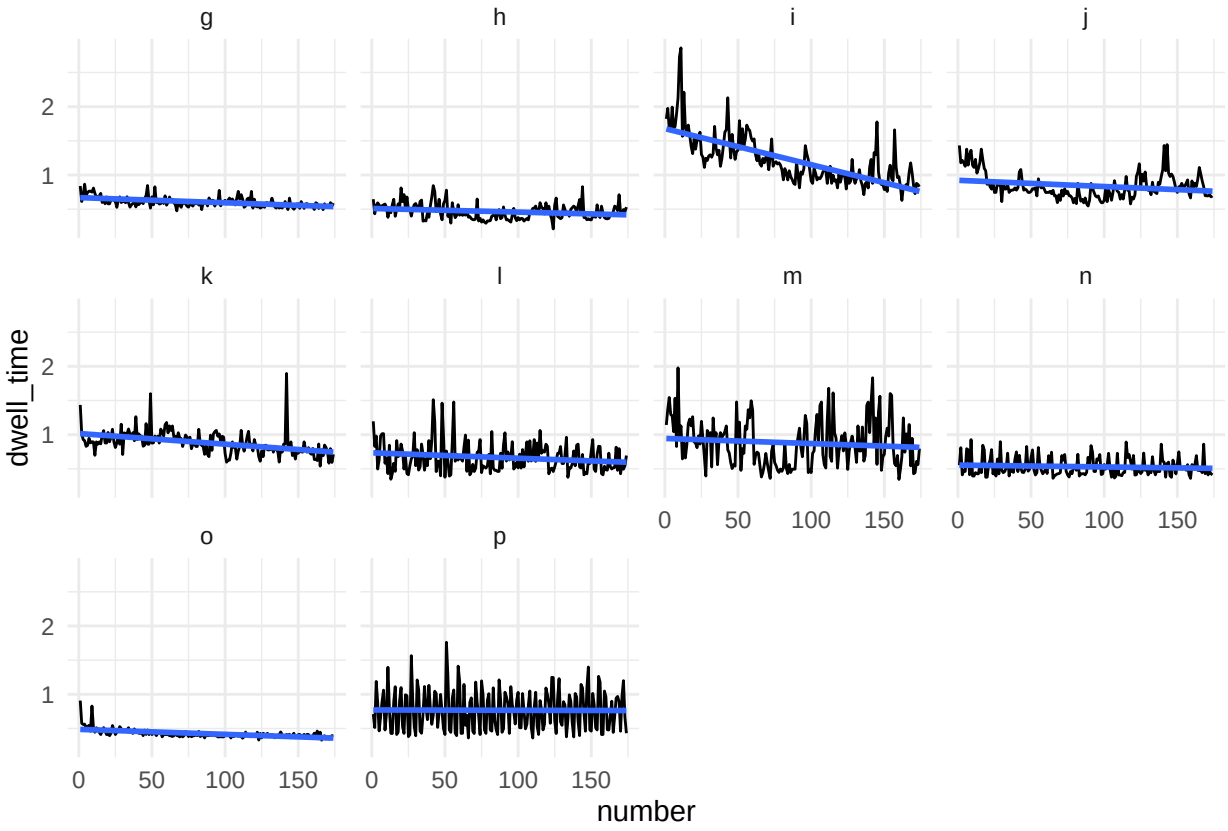

```
raw_A_long %>%
  ggplot(aes(x = number, y = dwell_time)) +
  facet_wrap("participant") +
  geom_line() +
  geom_smooth(method = "lm", se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
raw_B_long %>%
  ggplot(aes(x = number, y = dwell_time)) +
  facet_wrap("participant") +
  geom_line() +
  geom_smooth(method = "lm", se = F)
```

```
## `geom_smooth()` using formula 'y ~ x'
```


g
h
i
j
k
l
m
n
o
p
dwell_time
number
0
50
100
150
1
2