Making sense of Paramecium plasticity and growth across temperatures

Introduction

One goal of our project is to gain a better understanding of both the morphological and growth plasticity of paramecia in response to temperature. One reason for this is that the changes we see in paramecia under our warm and cold treatments are likely to be attributable partly to plasticity. Another reason is that plasticity is likely to play an important role in determining which outcrossed lines in our original evolution populations were likely to do well at high and low and temperatures. For example, some outcrossed lines may perform better than others at higher versus colder temperatures and vice versa. Moreover, there are some questions that we might be able to examine about plasticity in morphology and growth that are broadly interesting. For example, is the degree of plasticity in morphology related to the degree of plasticity in growth across temperatures? What changes in morphological traits are associated with changes in growth in paramecium?

What did we do?

The data we have to attempt to answer these questions comes from two experiments that we performed last year. The first was our TPC experiment for the 20 outcrossed lines that we used to start our original evolution populations. The second was a plasticity experiment in which we acclimated each of the 20 lines across different temperatures and then video phenotyped them to examine morphological and phenotypic plasticity across temperatures.

Characterizing Morphological and Behavioral Plasticity

The first thing we can do is try to characterize the morphological and behavioral plasticity of the paramecia aross temperatures. First, we will load the data and make some modifications to add some necessary information that is currently in the video file names. We will also drop the measurements from the highest temperature since very few cells survived at the highest temperature over the acclimation period.

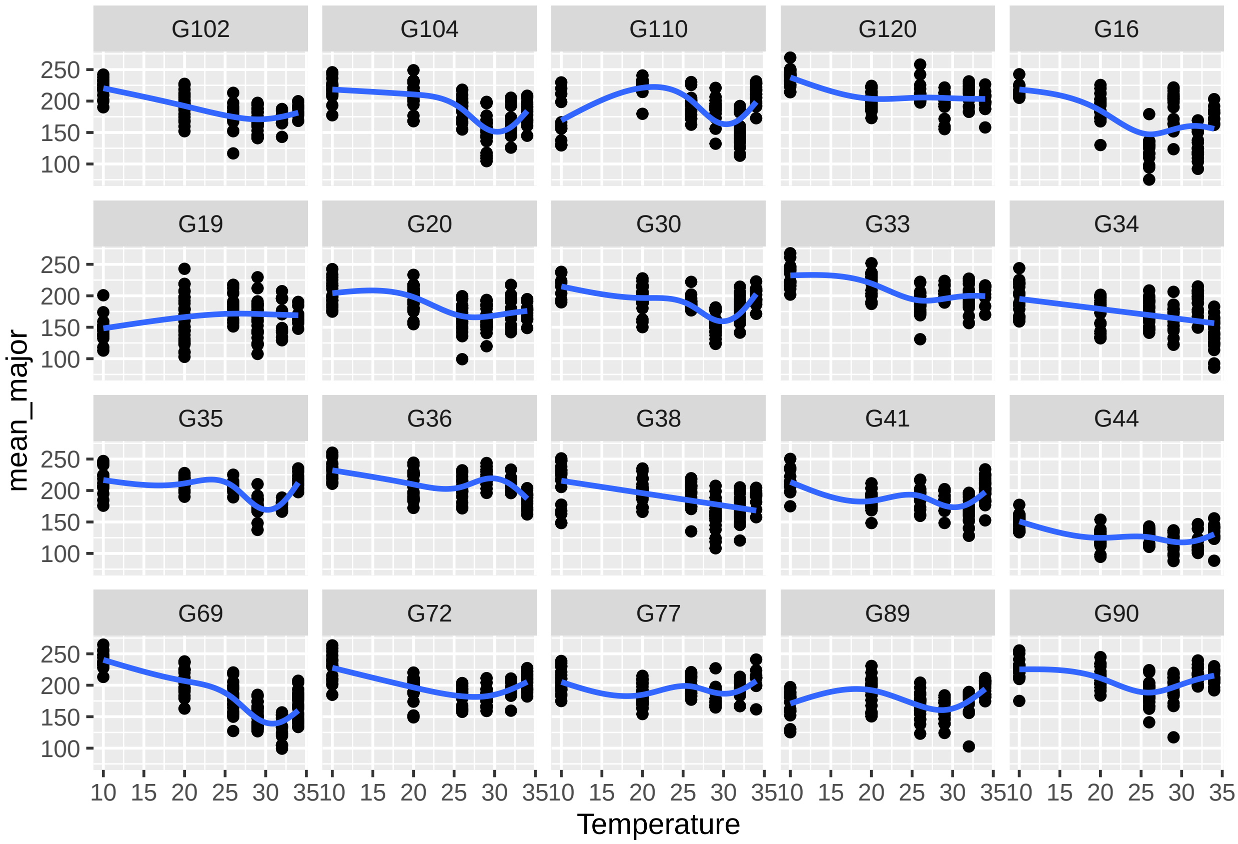
```
morph_data <- read.csv('Plast_StartPop_Data.csv')

morph_data <- morph_data %>% mutate(Line = as.factor(sapply(strsplit(morph_data$file, split = "_"),'[['
                                     Temperature = as.numeric(sapply(strsplit(morph_data$file, split =
                                    LineTemp = paste(Line, Temperature, sep = "_"))

morph_data <- filter(morph_data, Temperature < 37)
```

Now we can look at various morphological and behavioral traits and characterize their plasticity with temperature (we will focus on length, width, aspect ratio, and speed). To do so, I will use Generalized Additive Models (GAMs) as these models are capable of fitting flexible, nonlinear relationships when the forms of these relationships are not necessarily known *a priori*.

Length

First, we can look at a plot of paramecium length across temperatures.

```
ggplot(data = morph_data, aes(x = Temperature, y = mean_major)) + geom_point() + geom_smooth(method = '
  facet_wrap(facets = "Line", nrow = 4, ncol = 5)
```



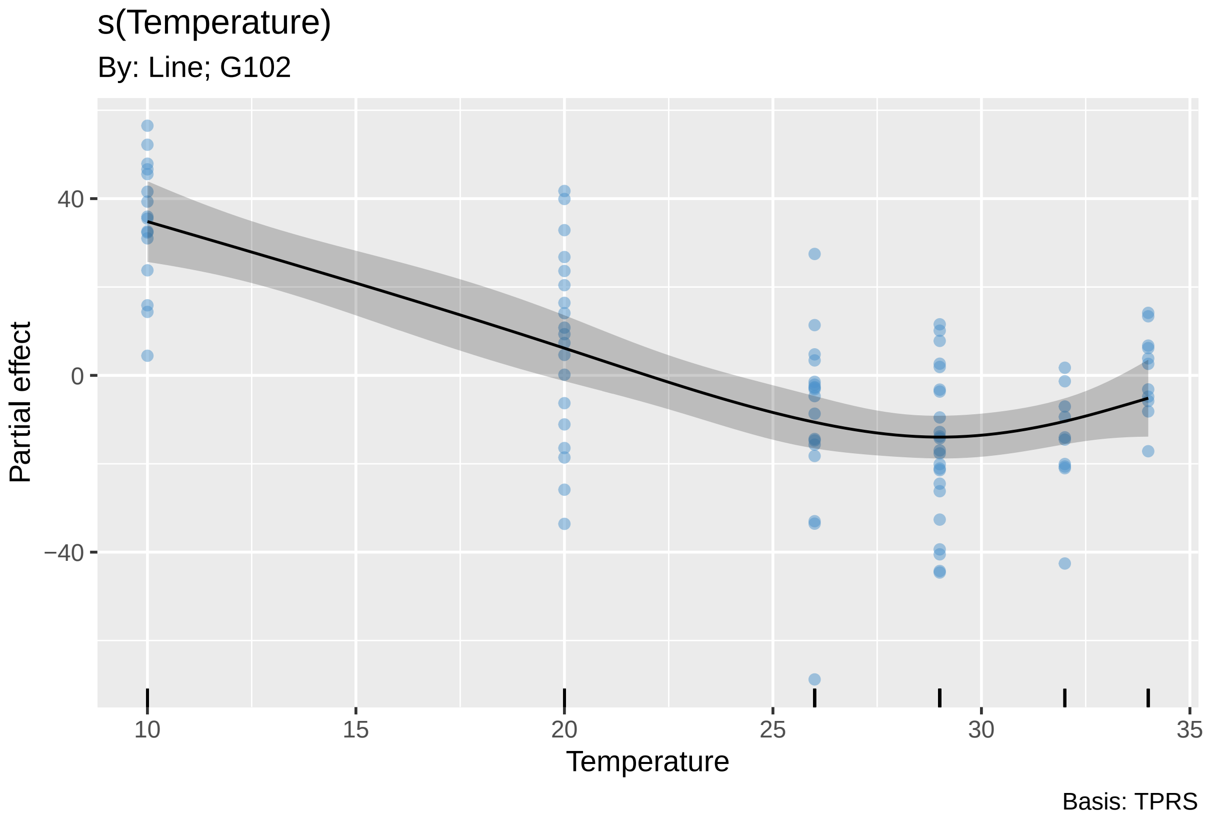
There is a lot going on, but, what we see generally is a tendency towards smaller lengths (mean major axis) as temperature increases with a slight increase in length at the highest temperature. Now we can run the formal GAMs and look at the resulting summary and plots.

```
gam_length <- gam(formula = mean_major ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = mor

summary(gam_length)
```

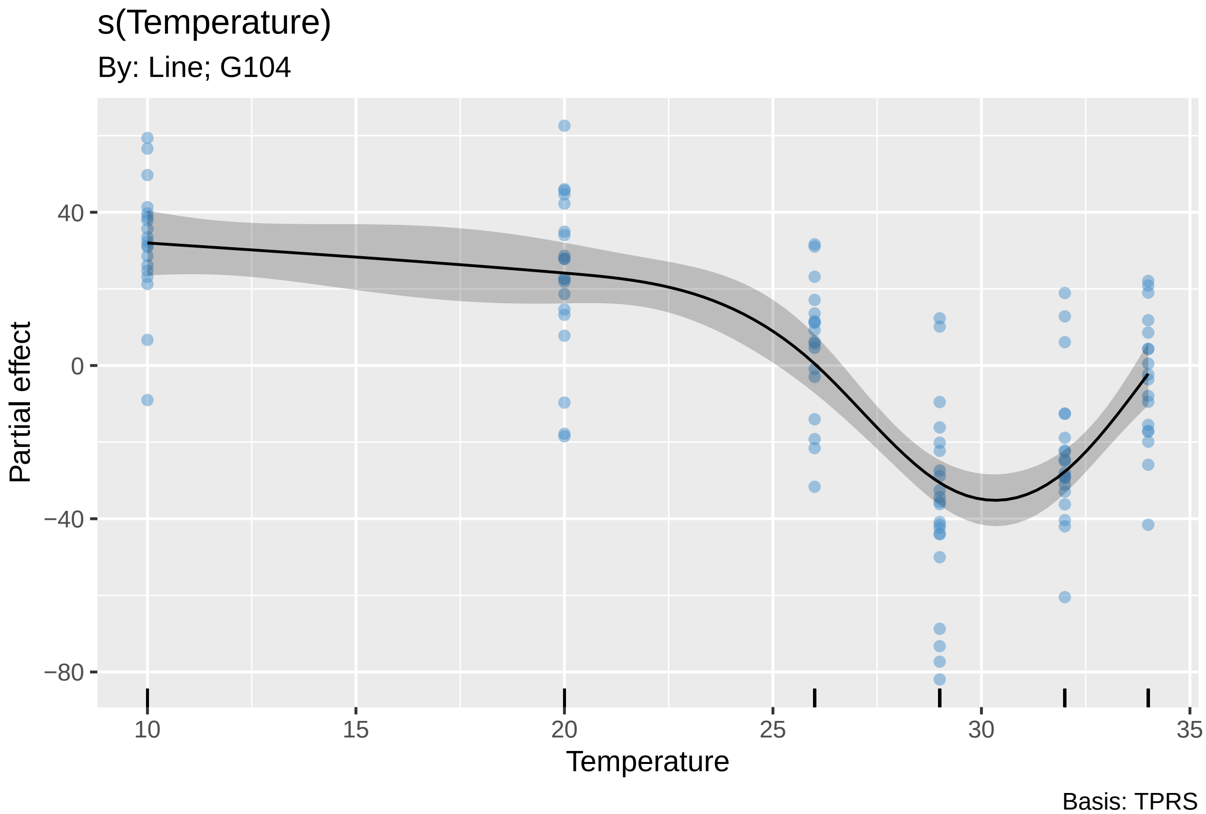
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean_major ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  185.632      2.090  88.810  < 2e-16 ***
## LineG104       0.775      2.821   0.275 0.783599
## LineG110       3.720      3.030   1.228 0.219565
## LineG120      23.846      2.977   8.011 1.91e-15 ***
## LineG16      -15.877      3.064  -5.181 2.42e-07 ***
## LineG19      -19.532      2.857  -6.837 1.07e-11 ***
## LineG20       -4.632      2.779  -1.667 0.095670 .
## LineG30        2.630      2.916   0.902 0.367243
## LineG33       20.876      2.947   7.084 1.93e-12 ***
## LineG34      -13.682      2.865  -4.775 1.93e-06 ***
```

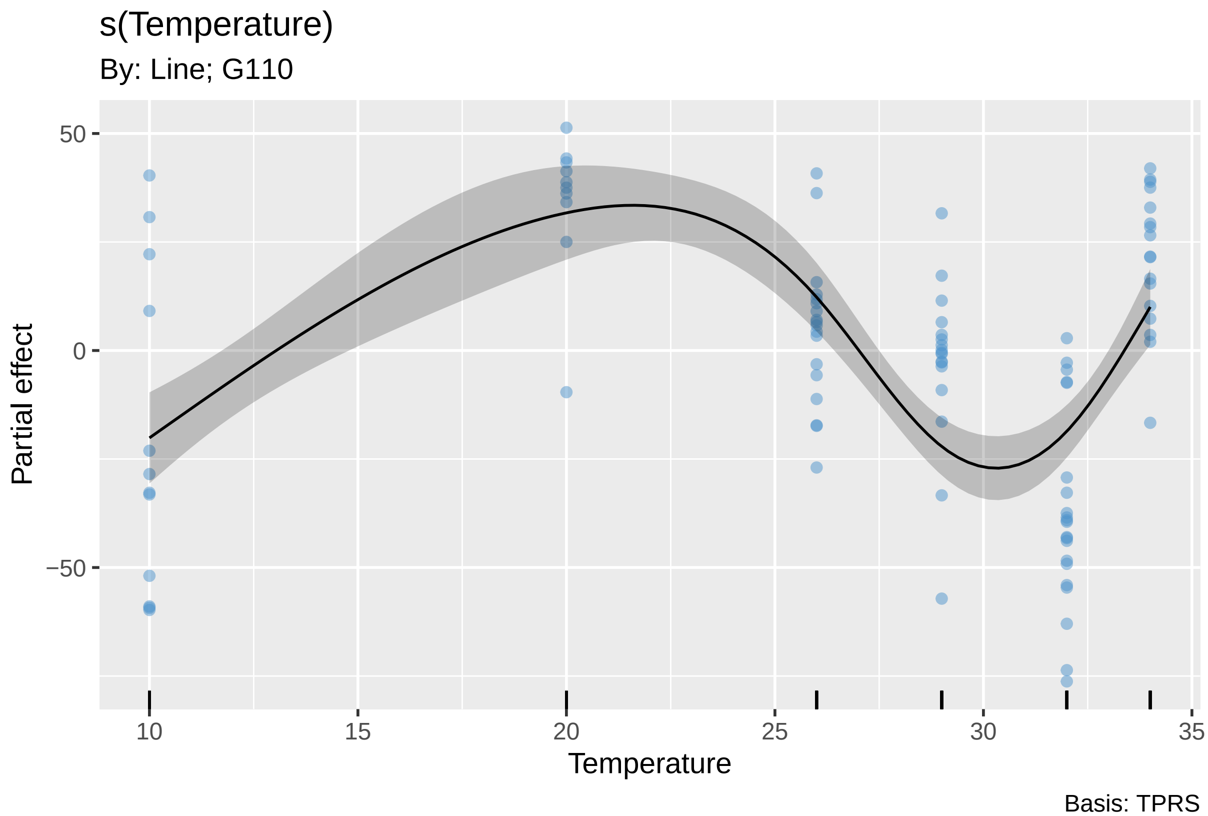
```
## LineG35       14.506      3.140   4.619 4.09e-06 ***
## LineG36       24.990      3.025   8.262 2.57e-16 ***
## LineG38        2.130      2.831   0.752 0.451844
## LineG41        4.123      2.917   1.413 0.157705
## LineG44      -57.476      2.931 -19.608  < 2e-16 ***
## LineG69       -5.517      2.822  -1.955 0.050686 .
## LineG72       11.708      2.835   4.130 3.78e-05 ***
## LineG77        9.696      2.924   3.317 0.000927 ***
## LineG89       -9.209      2.886  -3.191 0.001440 **
## LineG90       21.379      2.888   7.403 1.95e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf Ref.df      F  p-value
## s(Temperature):LineG102 2.859  3.288 21.218  < 2e-16 ***
## s(Temperature):LineG104 3.875  3.988 40.126  < 2e-16 ***
## s(Temperature):LineG110 3.891  3.992 20.379  < 2e-16 ***
## s(Temperature):LineG120 2.661  3.076 16.394  < 2e-16 ***
## s(Temperature):LineG16  3.825  3.977 29.403  < 2e-16 ***
## s(Temperature):LineG19  2.759  3.185  6.491 0.000149 ***
## s(Temperature):LineG20  3.424  3.771 17.545  < 2e-16 ***
## s(Temperature):LineG30  3.871  3.987 20.449  < 2e-16 ***
## s(Temperature):LineG33  3.277  3.670 13.263  < 2e-16 ***
## s(Temperature):LineG34  3.587  3.878 14.975  < 2e-16 ***
## s(Temperature):LineG35  3.855  3.984 14.597  < 2e-16 ***
## s(Temperature):LineG36  3.648  3.907  9.279 4.68e-07 ***
## s(Temperature):LineG38  3.847  3.983 22.332  < 2e-16 ***
## s(Temperature):LineG41  3.764  3.959  8.848 1.57e-06 ***
## s(Temperature):LineG44  2.436  2.863 10.881 1.66e-06 ***
## s(Temperature):LineG69  3.853  3.984 76.851  < 2e-16 ***
## s(Temperature):LineG72  3.151  3.547 18.806  < 2e-16 ***
## s(Temperature):LineG77  3.542  3.845  4.721 0.001999 **
## s(Temperature):LineG89  3.629  3.900 11.754  < 2e-16 ***
## s(Temperature):LineG90  3.423  3.780 10.195 4.90e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.598   Deviance explained = 61.5%
## -REML = 9312.9  Scale est. = 415.12    n = 2100
```

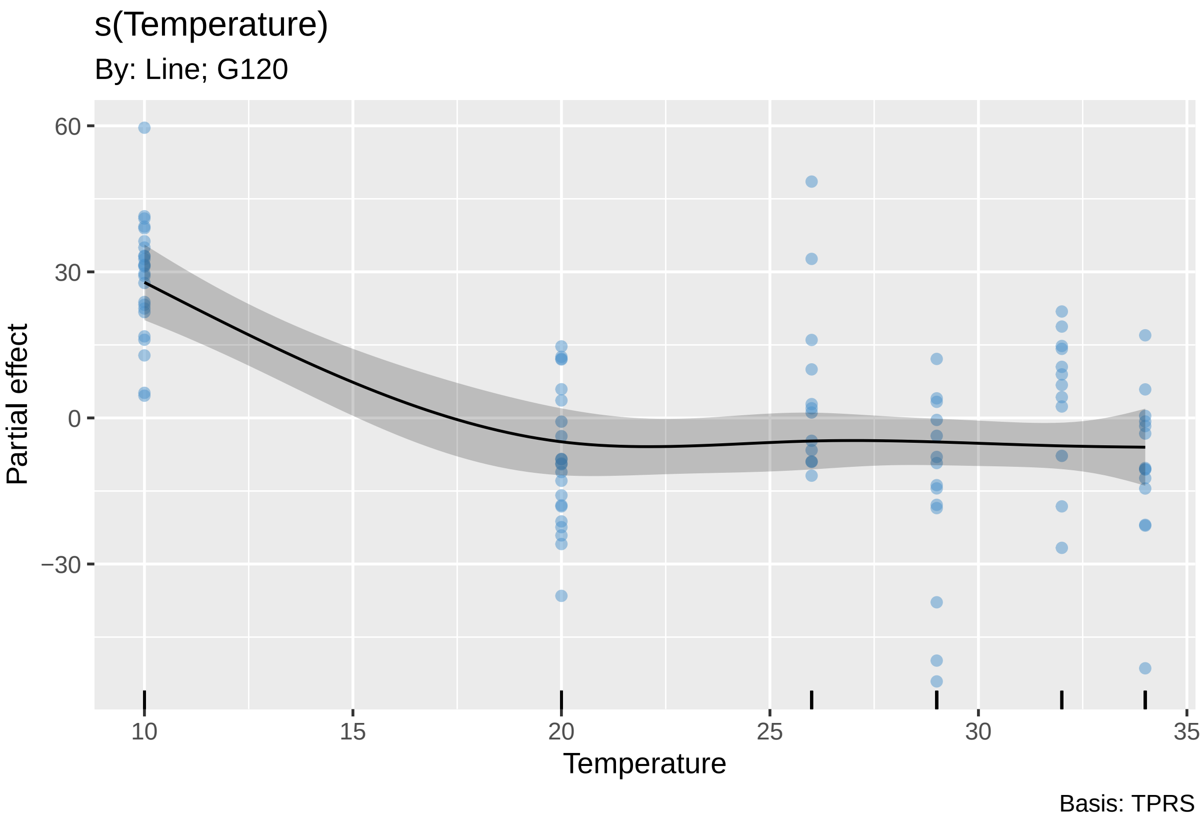
```r
for (i in 1:length(smooths(gam_length))) {
  out <- draw(gam_length, select = i, residuals = TRUE)
  print(out) }
```

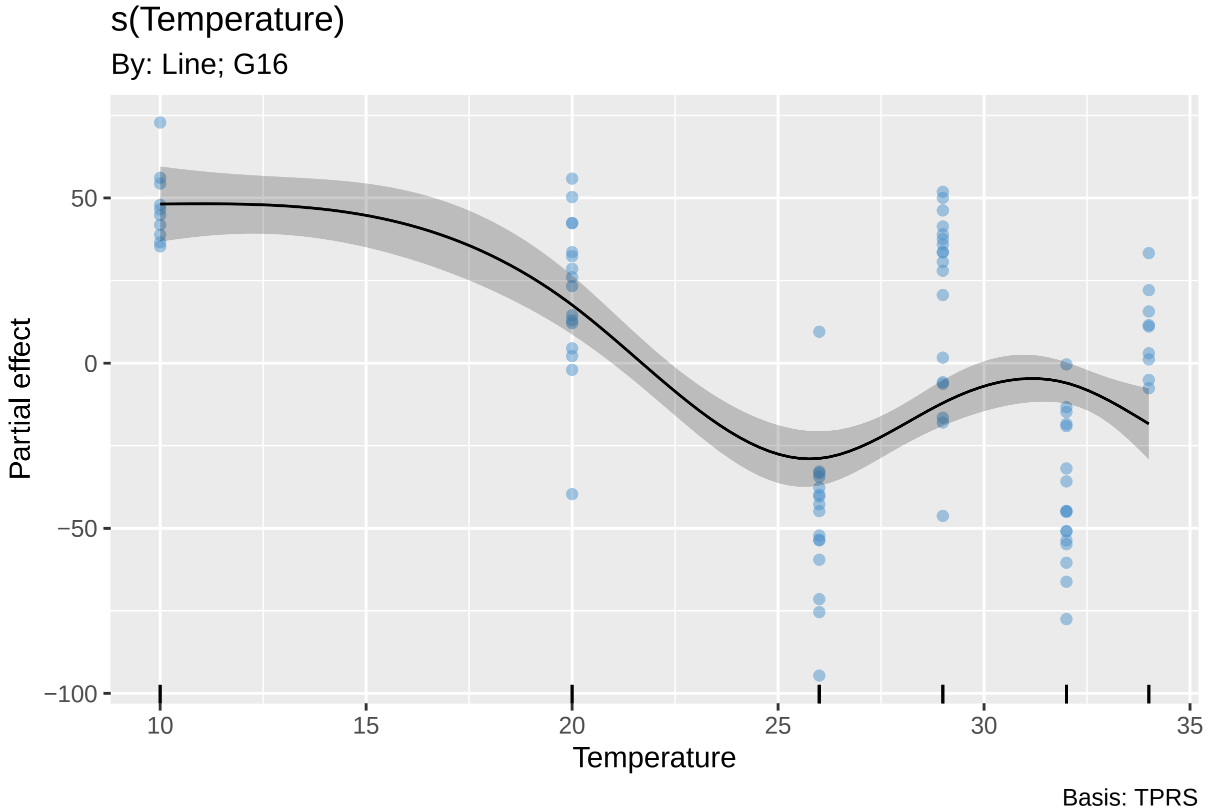


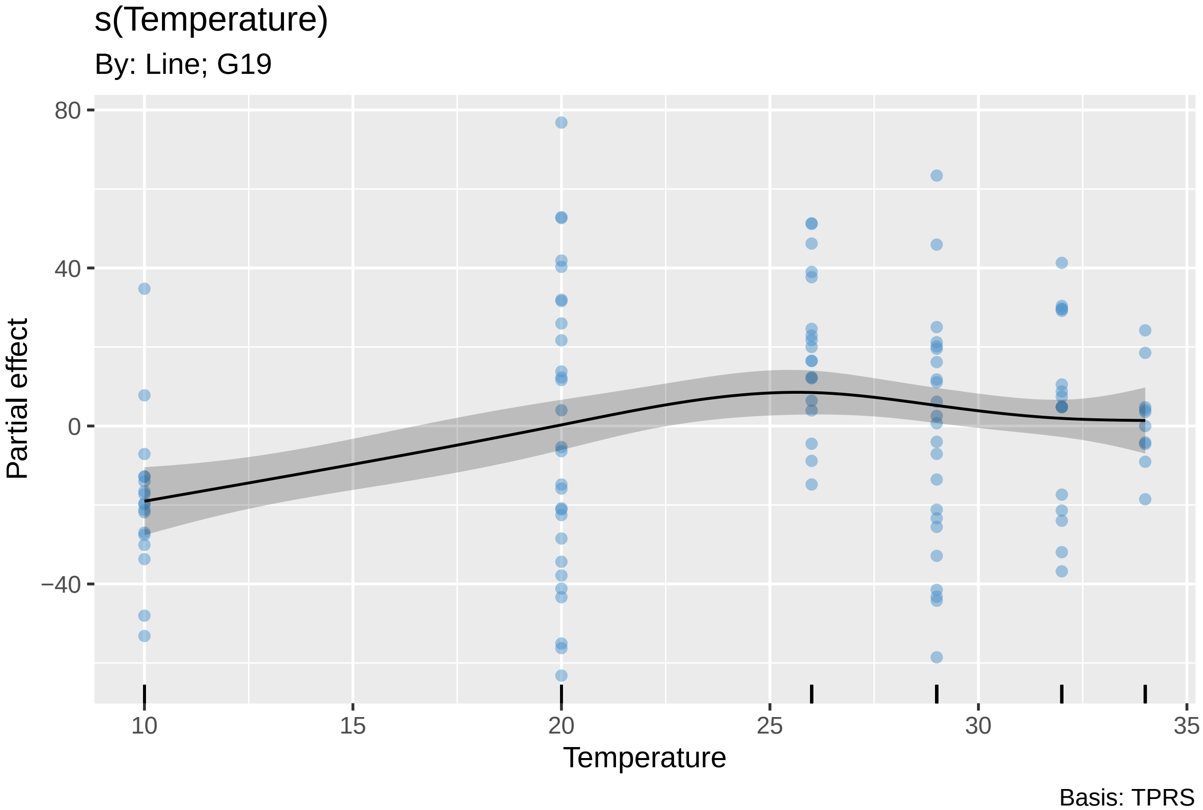
s(Temperature)
By: Line; G102
Partial effect
40
0
−40
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G104
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS

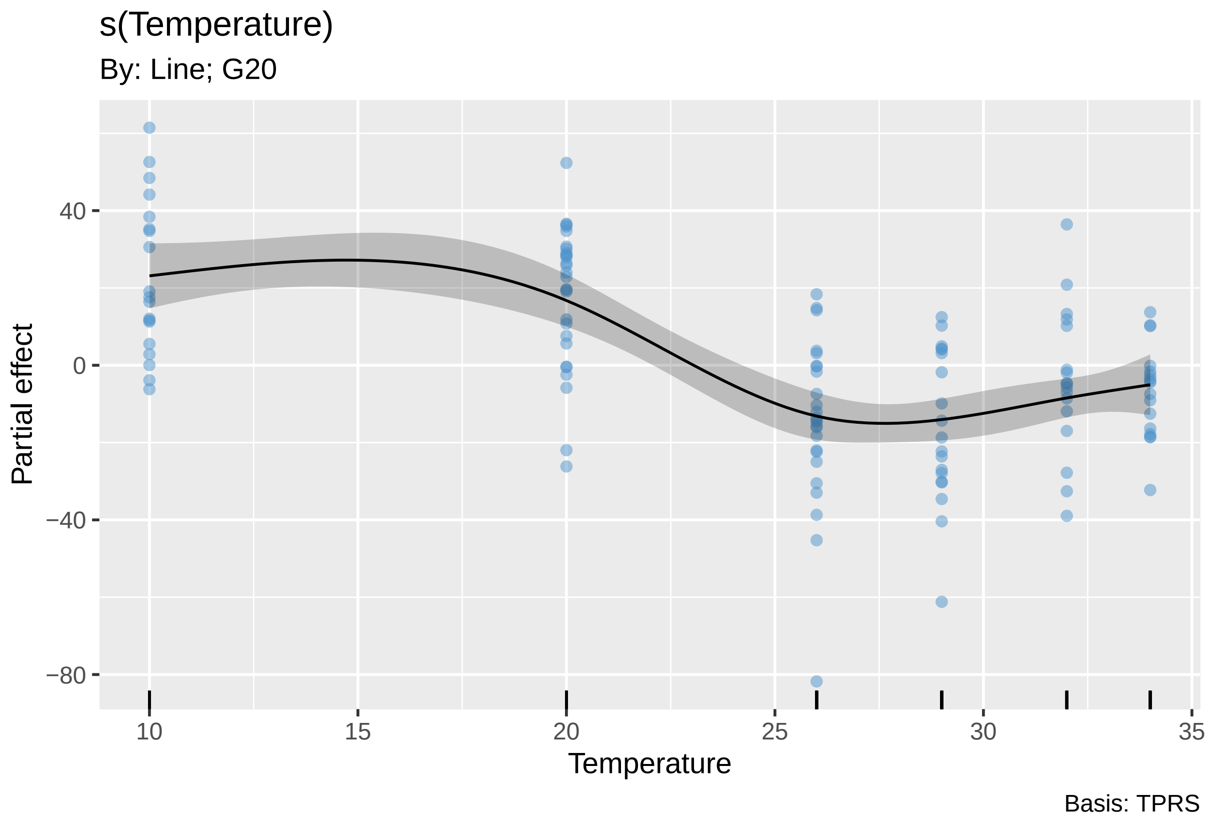

s(Temperature)
By: Line; G110
Partial effect
50
0
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

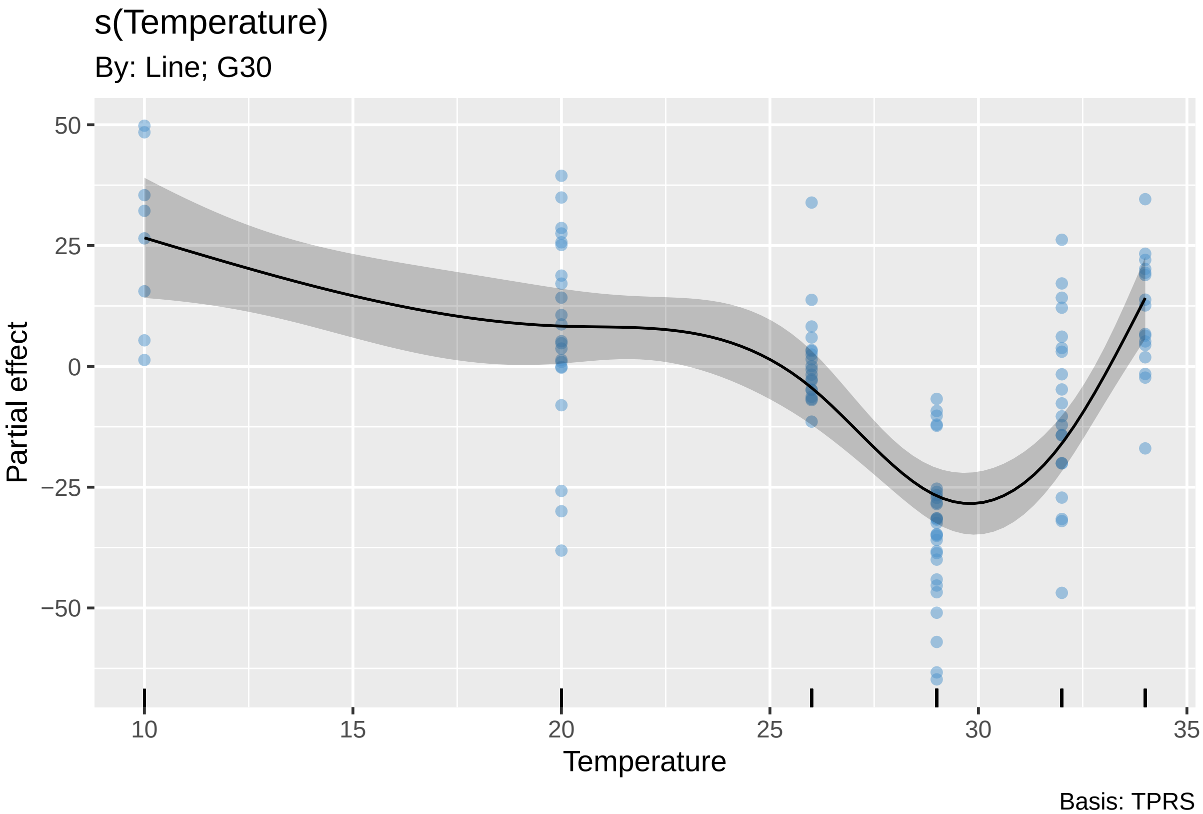

s(Temperature)
By: Line; G120
60
30
0
−30
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G16
Partial effect
50
0
−50
−100
10
15
20
25
30
35
Temperature
Basis: TPRS

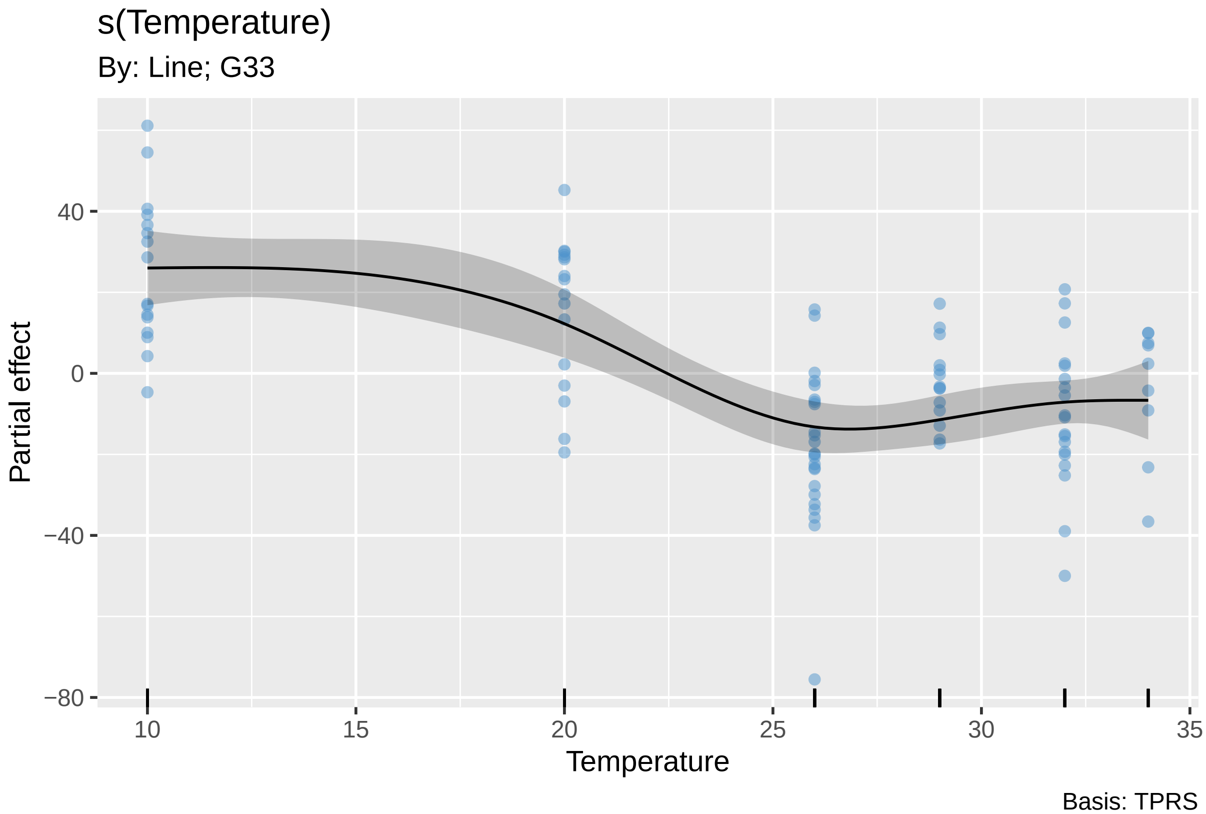


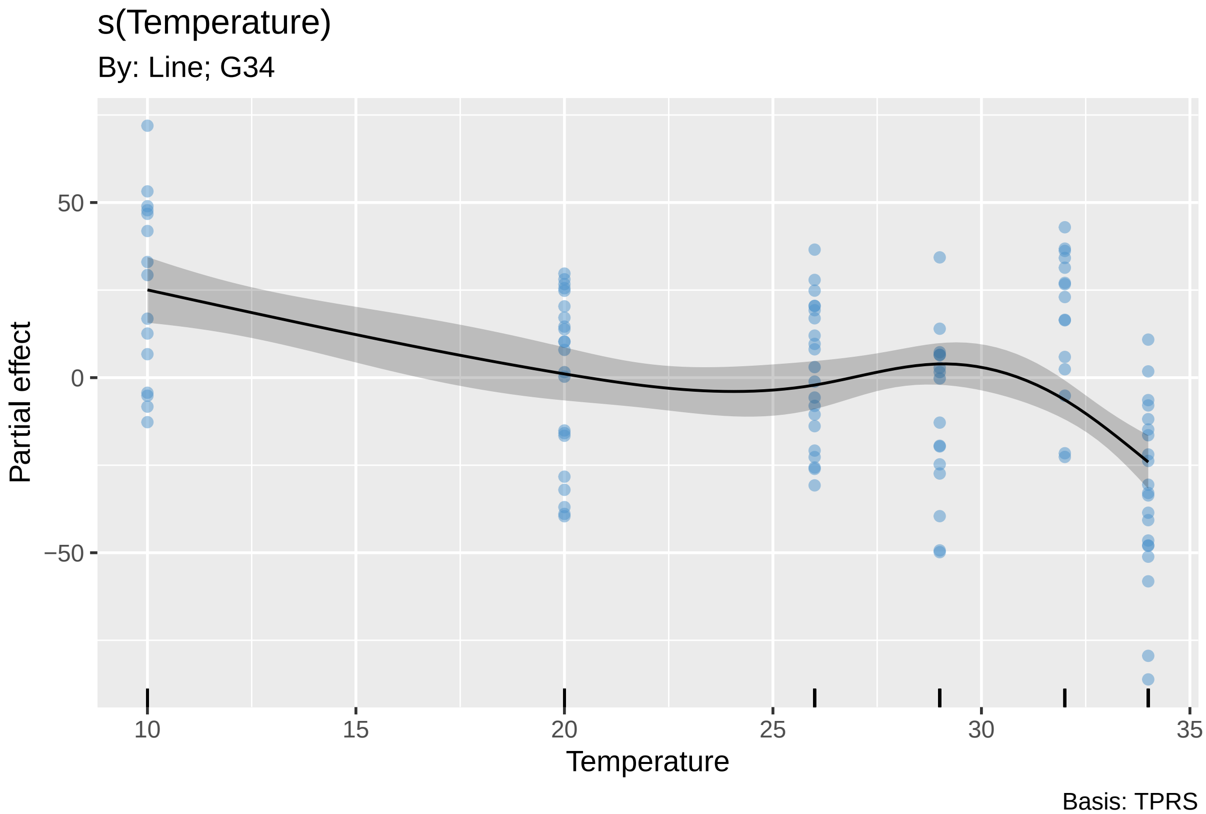
s(Temperature)
By: Line; G19
Partial effect
80
40
0
−40
10
15
20
25
30
35
Temperature
Basis: TPRS

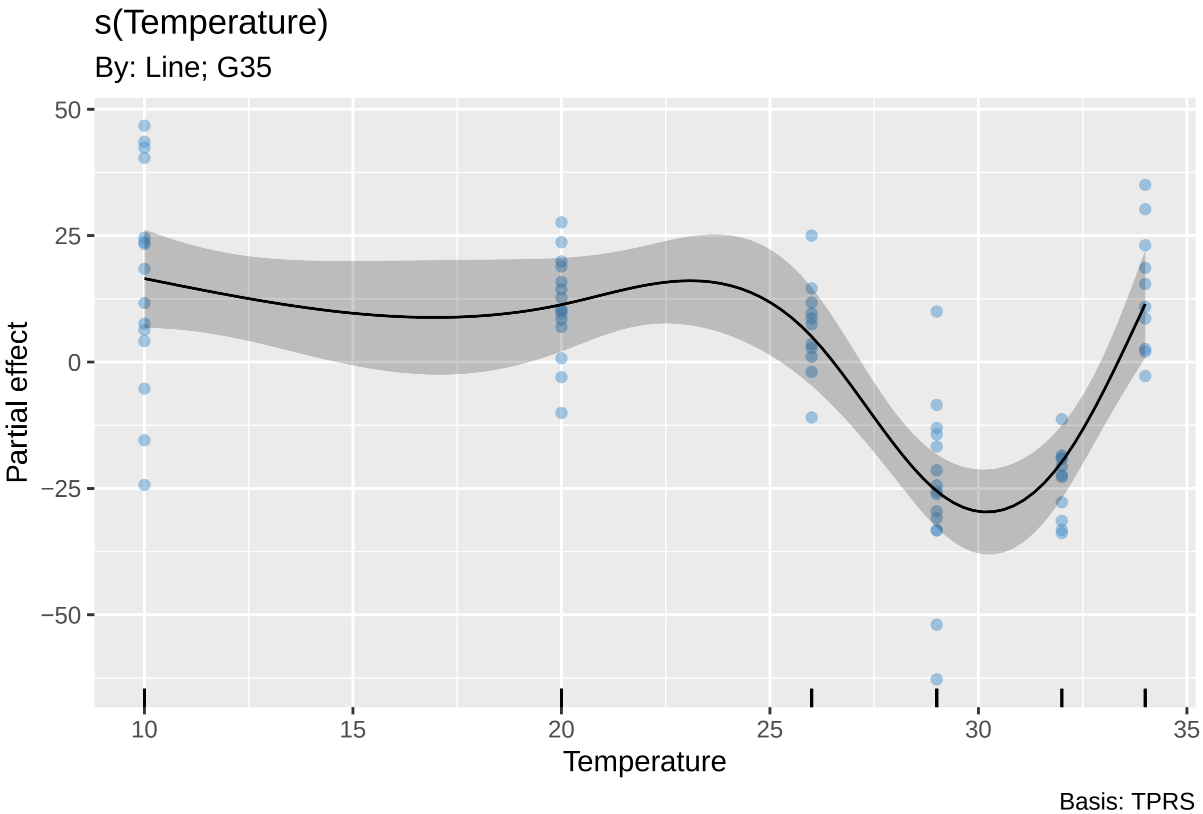

s(Temperature)
By: Line; G20
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS

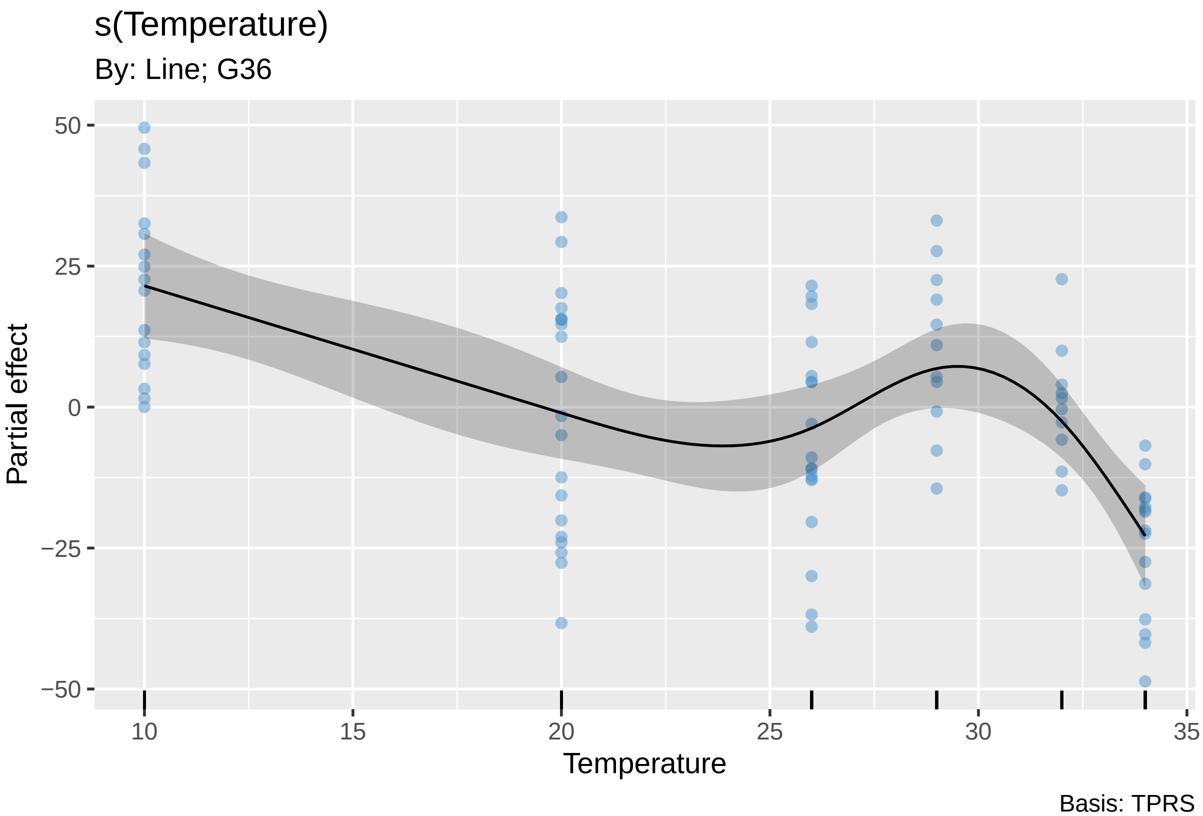


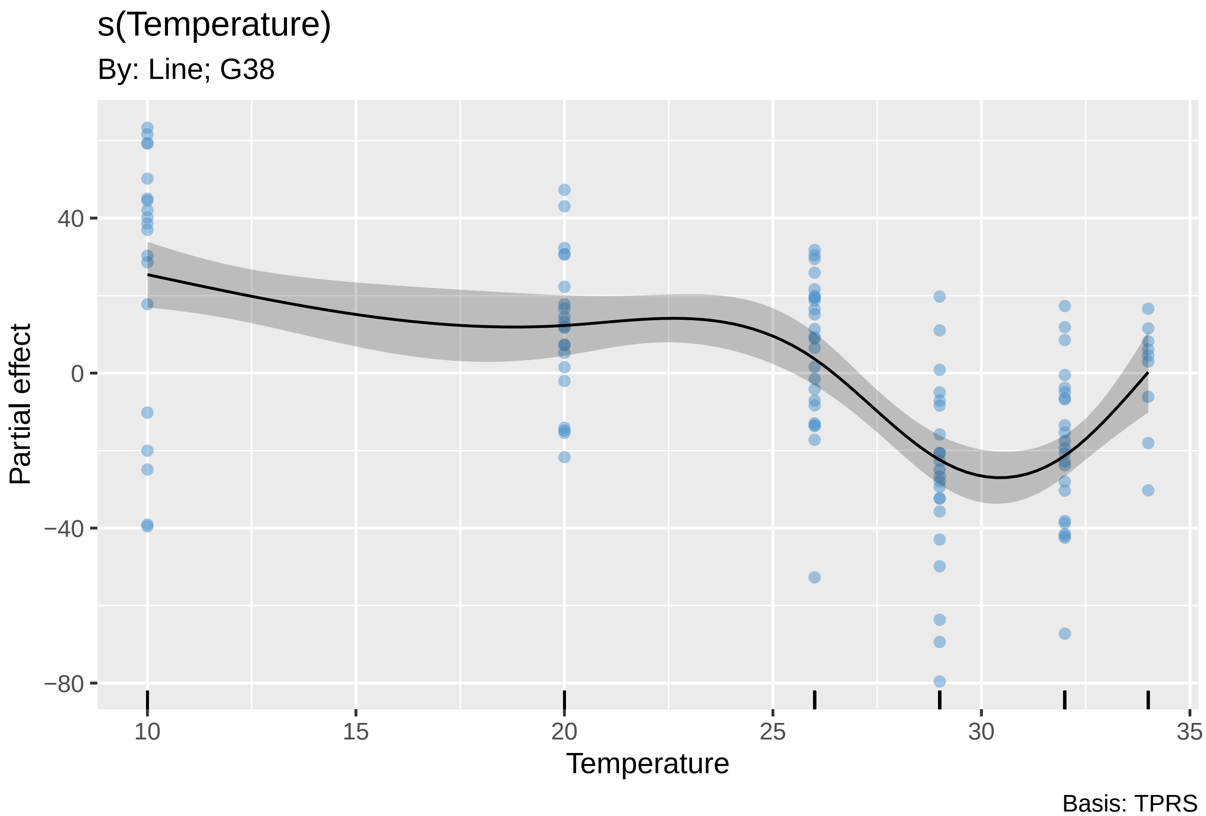
s(Temperature)
By: Line; G30
Partial effect
50
25
0
−25
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

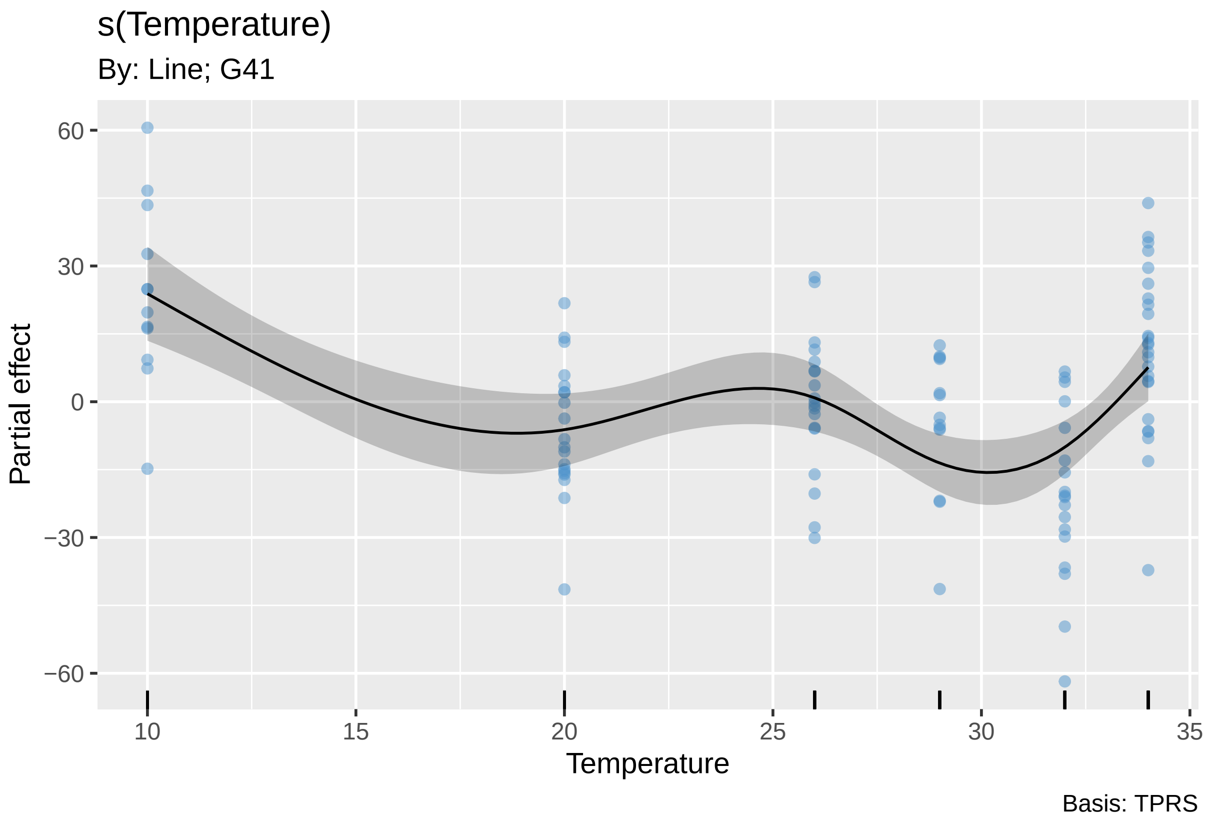

s(Temperature)
By: Line; G33
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS

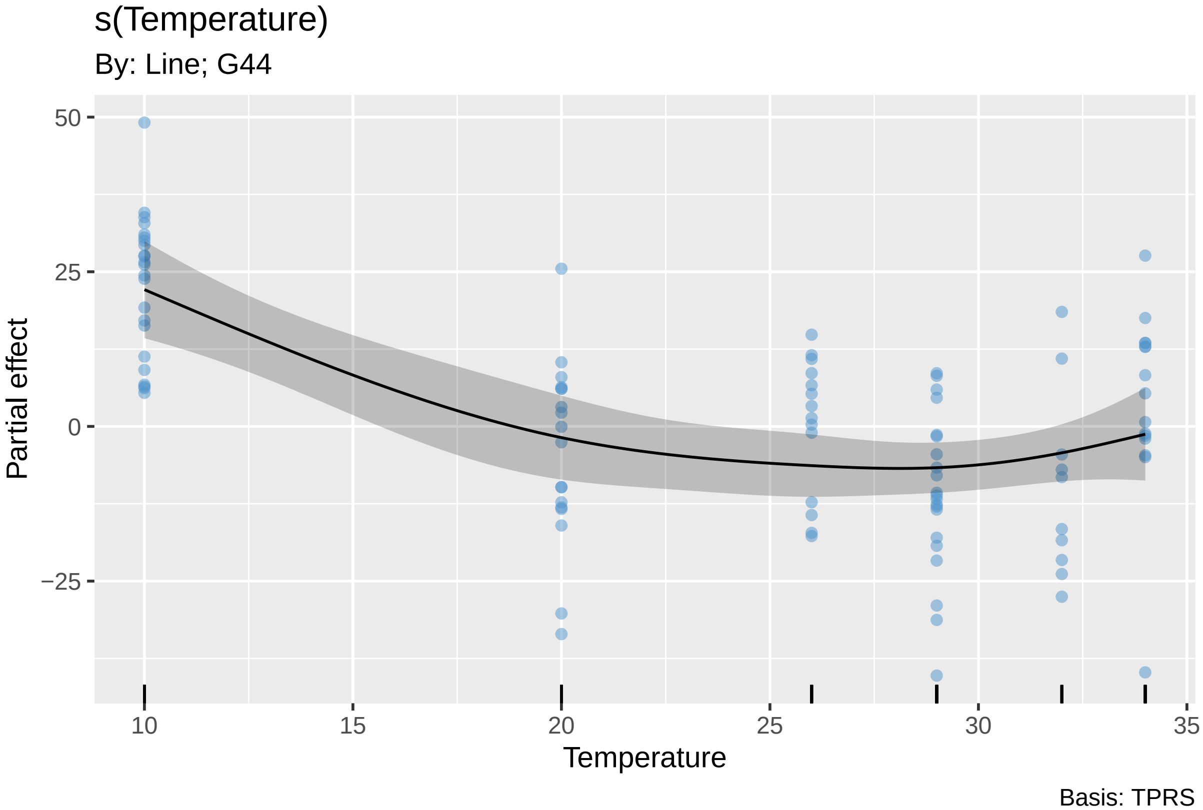

s(Temperature)
By: Line; G34
Partial effect
50
0
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

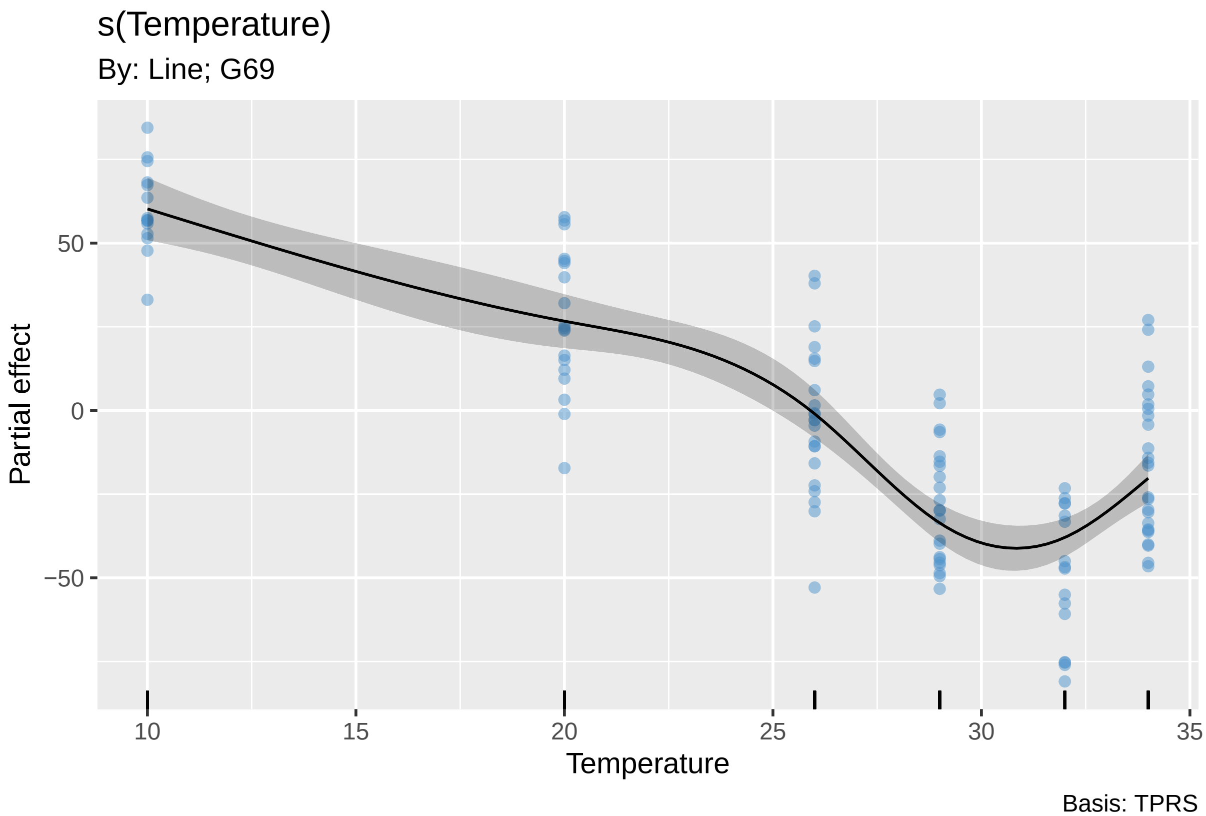

s(Temperature)
By: Line; G35
Partial effect
50
25
0
−25
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

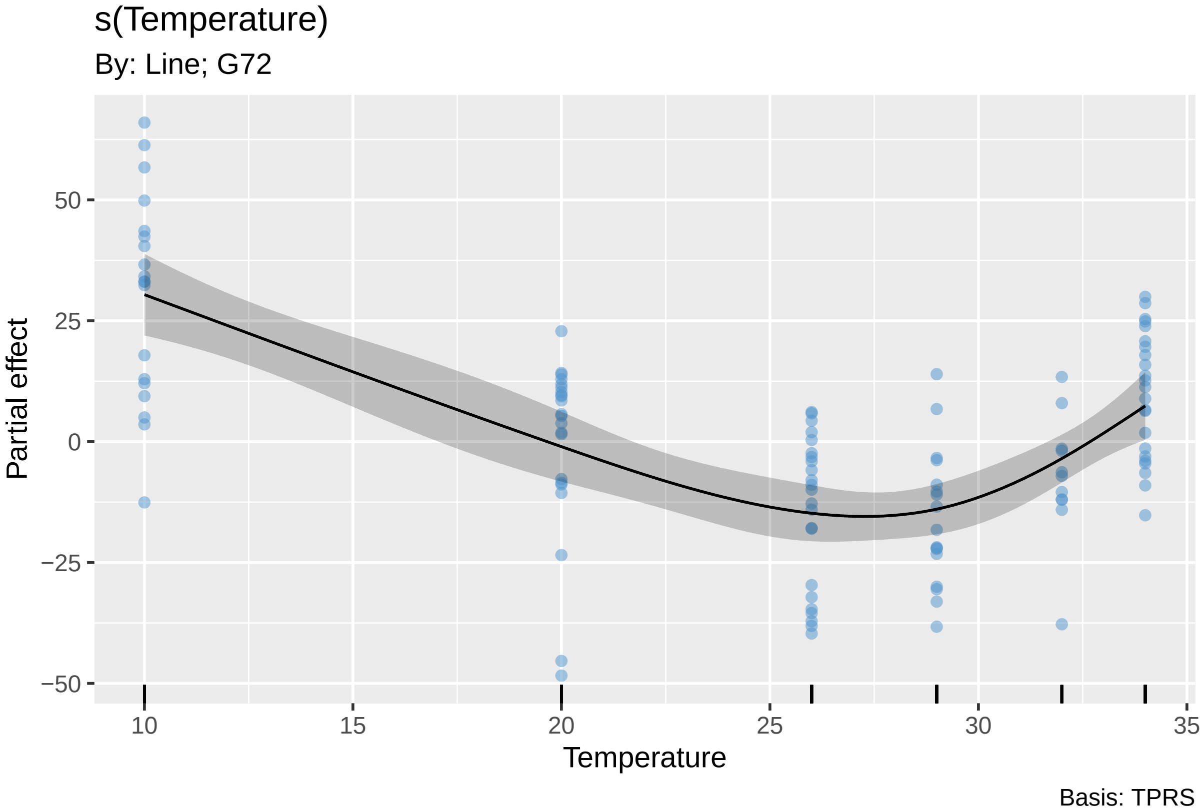

s(Temperature)
By: Line; G36
50
25
0
−25
−50
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

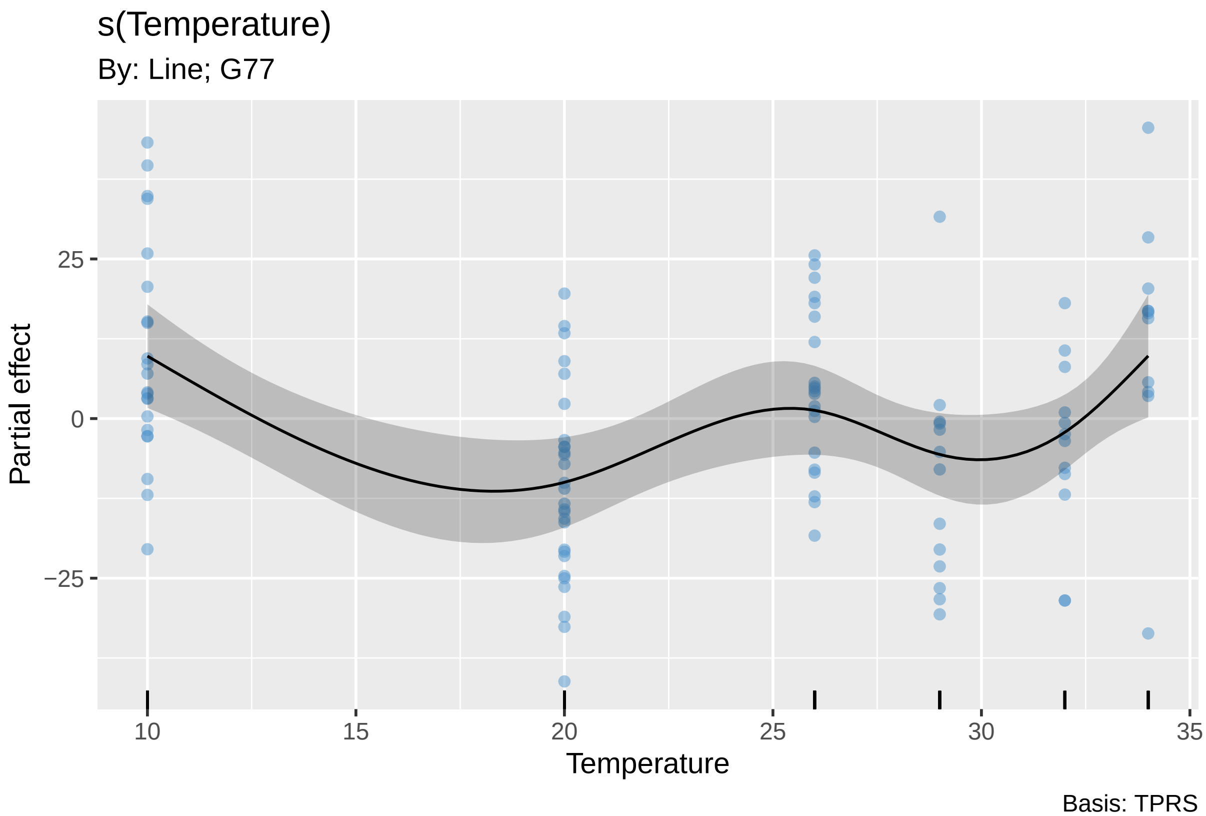

s(Temperature)
By: Line; G38
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS

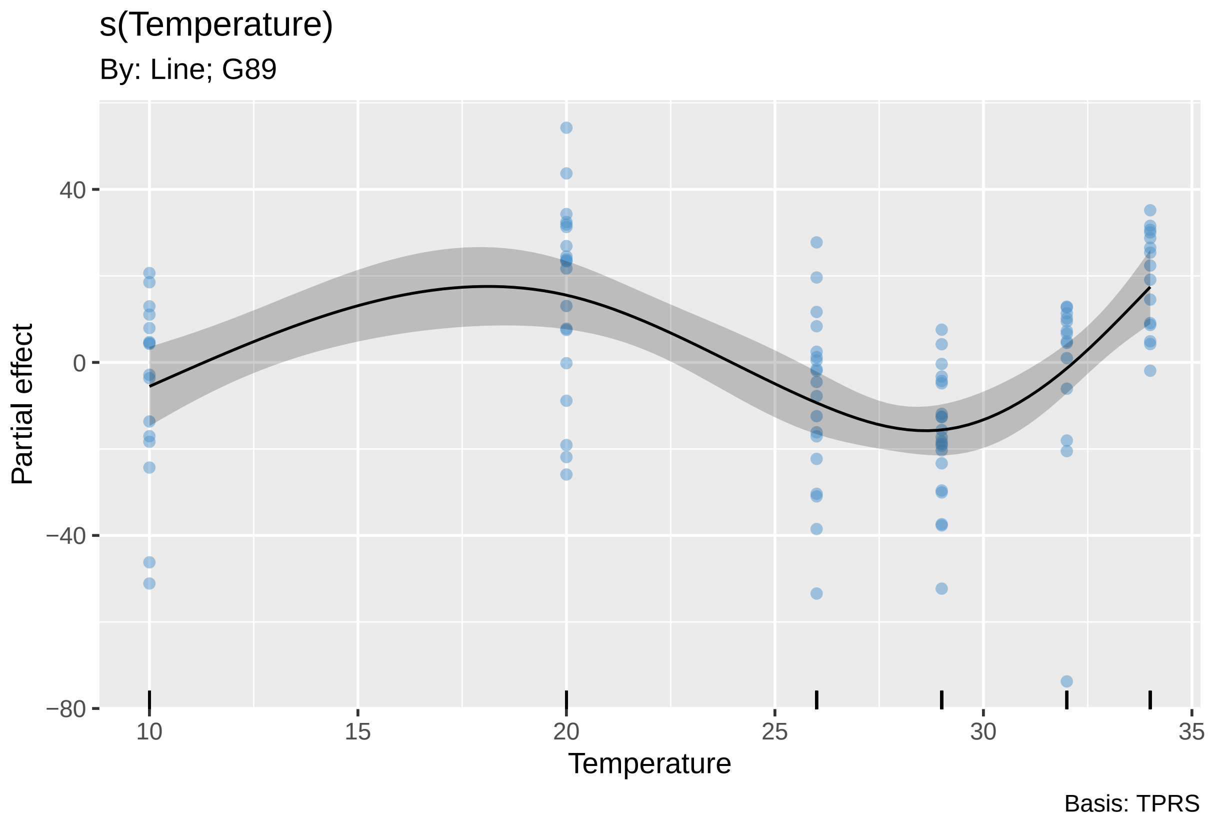

s(Temperature)
By: Line; G41
Partial effect
60
30
0
−30
−60
10
15
20
25
30
35
Temperature
Basis: TPRS

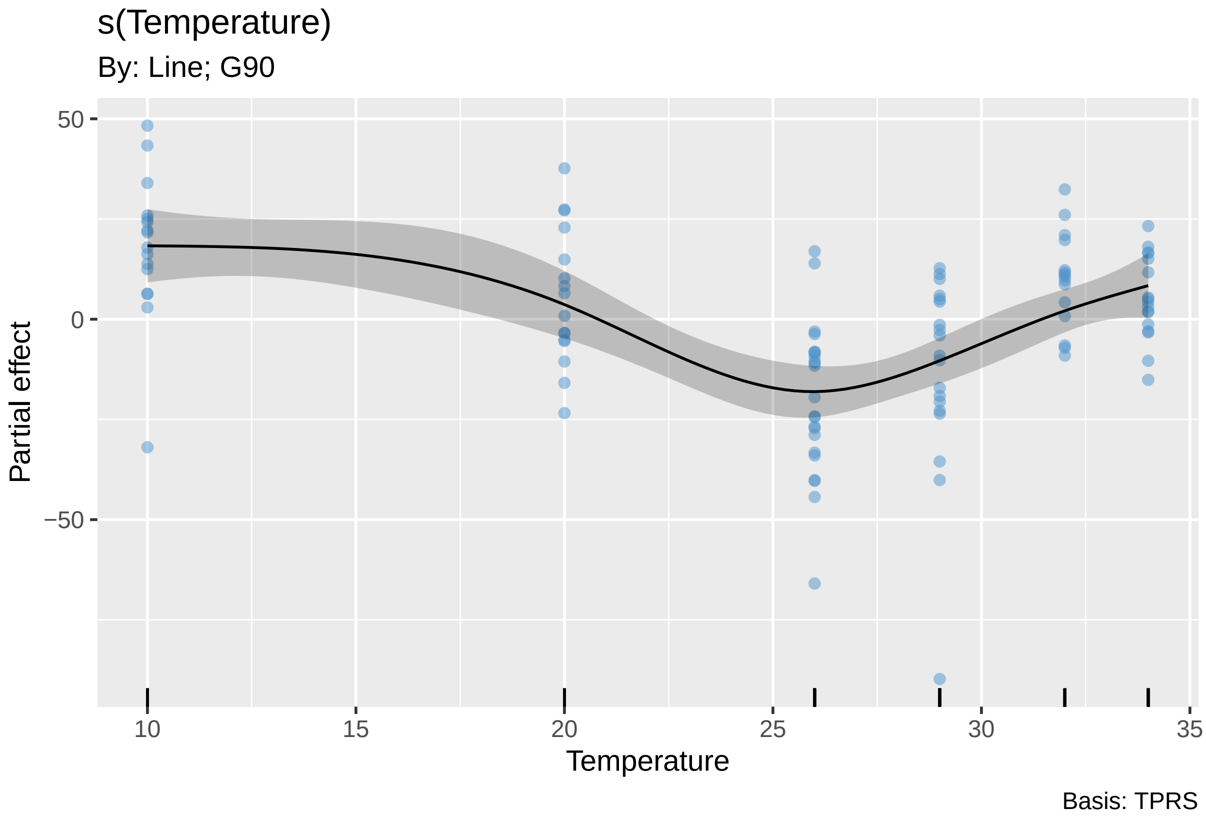

s(Temperature)
By: Line; G44
Partial effect
50
25
0
−25
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G69
Partial effect
50
0
−50
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G72
Partial effect
50
25
0
−25
−50
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G77
Partial effect
25
0
−25
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G89
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS



In general, these fits look pretty good. We can also examine whether including the interaction between line and temperature (i.e. a GxE interaction) improves the fit of the model to the data. We will fit a model without the interaction and then compare the AIC values of the two models.

```
gam_length_noint <- gam(formula = mean_major ~ Line + s(Temperature, k = 5, bs = 'tp'), data = morph_da

AIC(gam_length, gam_length_noint)
```

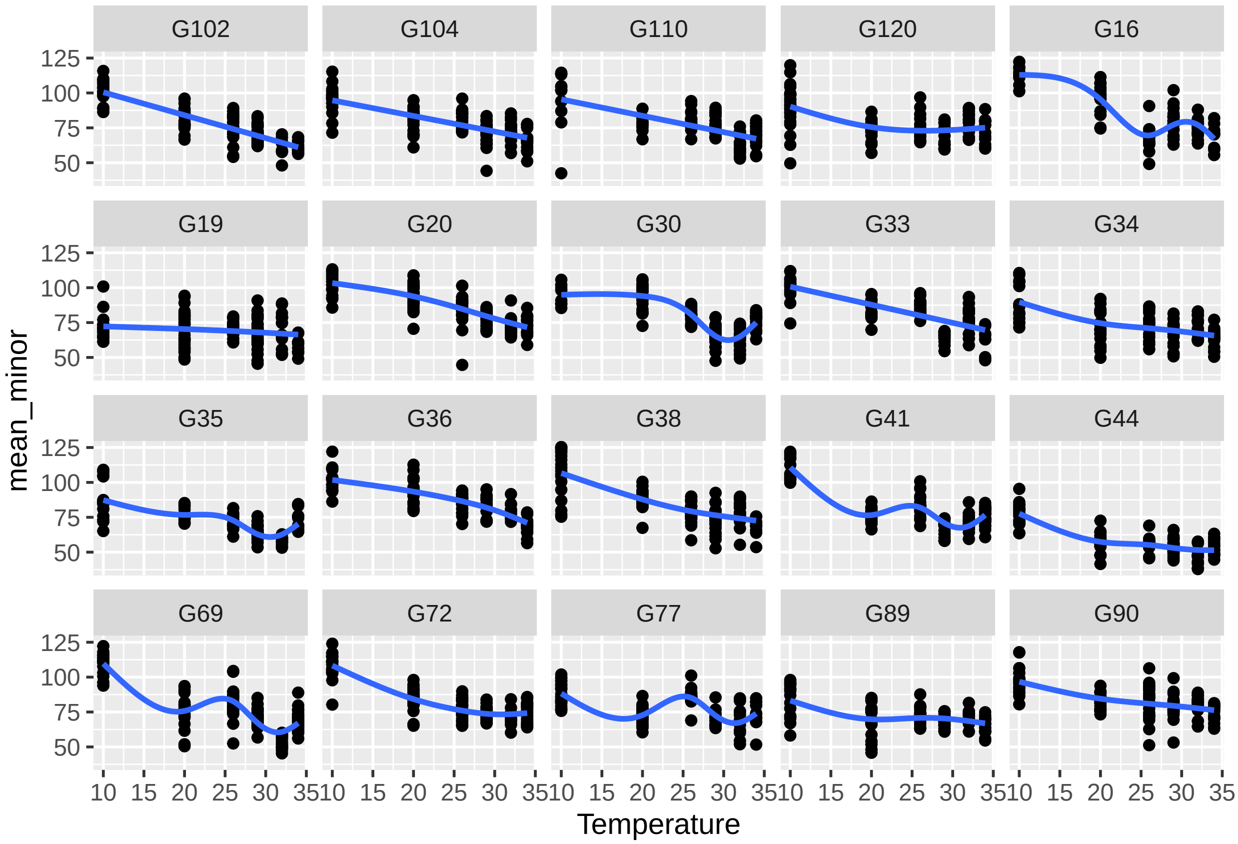
```
##                        df      AIC
## gam_length       95.56545 18719.53
## gam_length_noint 24.99633 19293.96
```

Overall, we see that the GAM including the interaction has a much, much lower AIC score telling us that including the interaction term increases the predictive ability of the model substantially (ΔAIC = 574.4378889).

Width

Next, we can look at paramecium width across temperatures.

```
ggplot(data = morph_data, aes(x = Temperature, y = mean_minor)) + geom_point() + geom_smooth(method = '
  facet_wrap(facets = "Line", nrow = 4, ncol = 5)
```



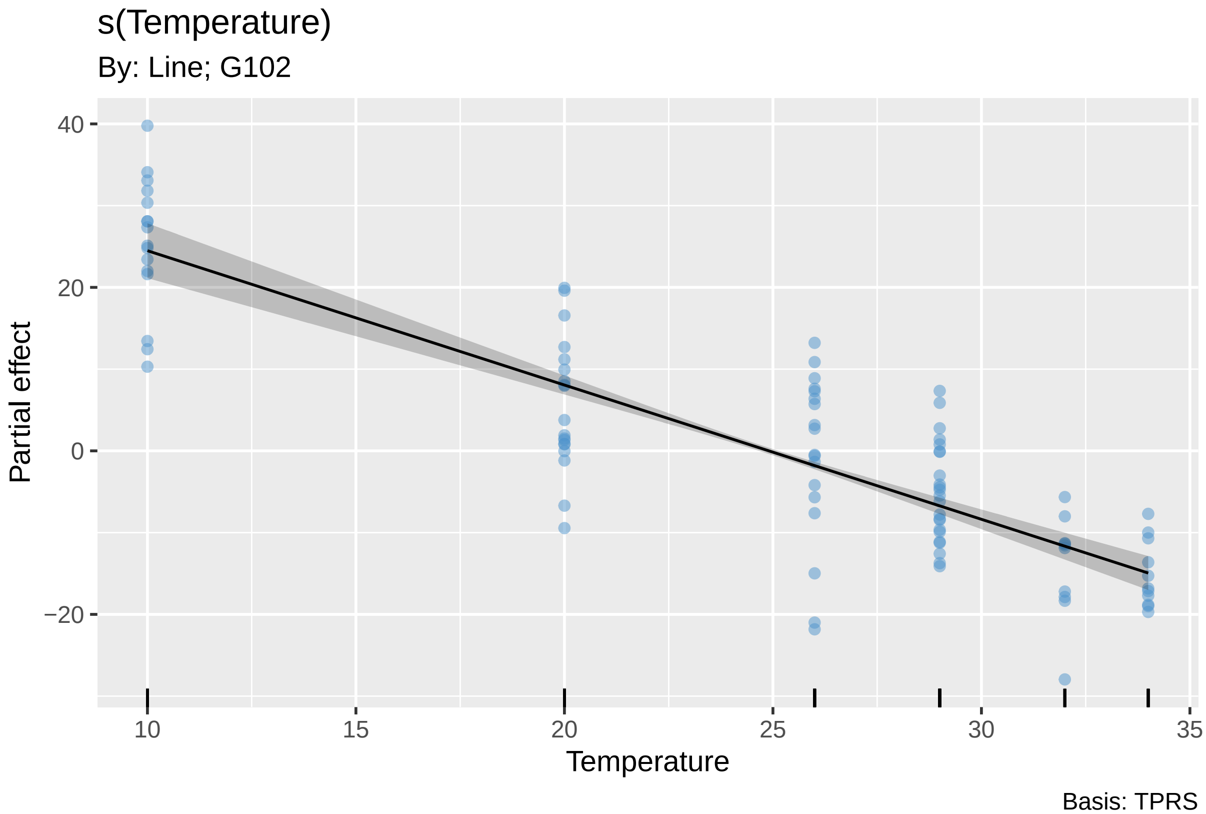
In general, we see a similar trend as the one we saw with length – paramecia are generally getting less wide as temperatures increase. Again, we will run GAMs and look at the results.

```
gam_width <- gam(formula = mean_minor ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = morph

summary(gam_width)
```

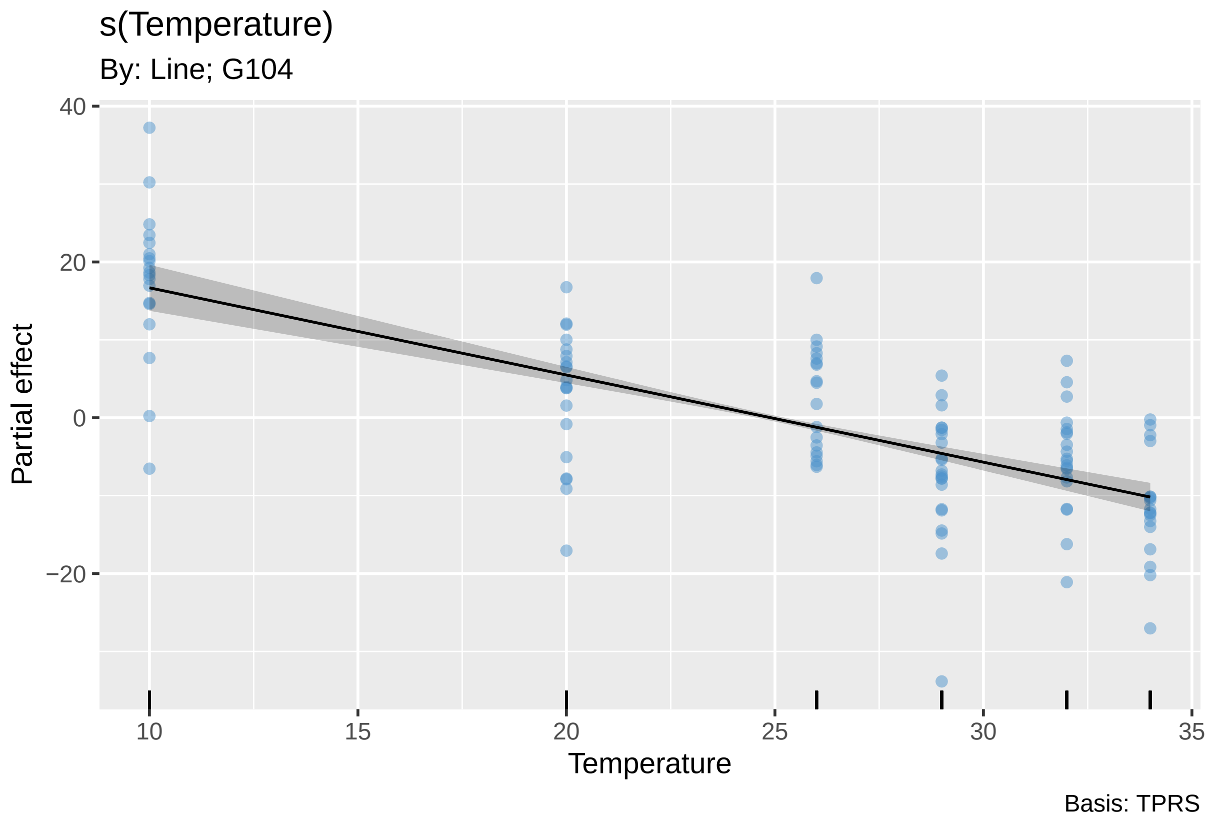
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean_minor ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   76.0081     0.8863  85.761  < 2e-16 ***
## LineG104       2.0314     1.1979   1.696 0.090092 .
## LineG110       1.8815     1.2756   1.475 0.140361
## LineG120       0.7053     1.2670   0.557 0.577819
## LineG16        7.4243     1.3055   5.687 1.48e-08 ***
## LineG19       -7.1552     1.2131  -5.898 4.30e-09 ***
## LineG20        9.0764     1.1818   7.680 2.46e-14 ***
## LineG30        3.6209     1.2417   2.916 0.003584 **
## LineG33        5.4163     1.2491   4.336 1.52e-05 ***
## LineG34       -3.0320     1.2174  -2.491 0.012834 *
## LineG35       -3.8116     1.3373  -2.850 0.004412 **
```

```
## LineG36       9.5426     1.2808   7.451 1.37e-13 ***
## LineG38       7.0089     1.1971   5.855 5.55e-09 ***
## LineG41       4.8101     1.2424   3.872 0.000111 ***
## LineG44     -18.3084     1.2482 -14.668  < 2e-16 ***
## LineG69       1.3793     1.2011   1.148 0.250955
## LineG72       5.6490     1.2051   4.688 2.95e-06 ***
## LineG77       0.8237     1.2452   0.662 0.508343
## LineG89      -4.4420     1.2285  -3.616 0.000307 ***
## LineG90       6.6714     1.2273   5.436 6.10e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                             edf Ref.df       F p-value
## s(Temperature):LineG102 1.002  1.004 205.871  <2e-16 ***
## s(Temperature):LineG104 1.002  1.004 124.573  <2e-16 ***
## s(Temperature):LineG110 1.002  1.004  97.609  <2e-16 ***
## s(Temperature):LineG120 2.542  2.959  22.335  <2e-16 ***
## s(Temperature):LineG16  3.888  3.991  58.909  <2e-16 ***
## s(Temperature):LineG19  1.767  2.152   3.014  0.0533 .
## s(Temperature):LineG20  2.148  2.569  71.307  <2e-16 ***
## s(Temperature):LineG30  3.883  3.990  52.943  <2e-16 ***
## s(Temperature):LineG33  1.008  1.016 128.257  <2e-16 ***
## s(Temperature):LineG34  2.544  2.983  27.725  <2e-16 ***
## s(Temperature):LineG35  3.692  3.929  21.300  <2e-16 ***
## s(Temperature):LineG36  2.311  2.731  47.210  <2e-16 ***
## s(Temperature):LineG38  2.305  2.741  73.861  <2e-16 ***
## s(Temperature):LineG41  3.878  3.989  50.986  <2e-16 ***
## s(Temperature):LineG44  2.661  3.086  44.178  <2e-16 ***
## s(Temperature):LineG69  3.928  3.996  84.186  <2e-16 ***
## s(Temperature):LineG72  2.791  3.216  71.661  <2e-16 ***
## s(Temperature):LineG77  3.885  3.990  21.156  <2e-16 ***
## s(Temperature):LineG89  2.859  3.290  11.461  <2e-16 ***
## s(Temperature):LineG90  2.209  2.636  22.521  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.647   Deviance explained = 65.9%
## -REML = 7527.7  Scale est. = 75.899    n = 2100
```

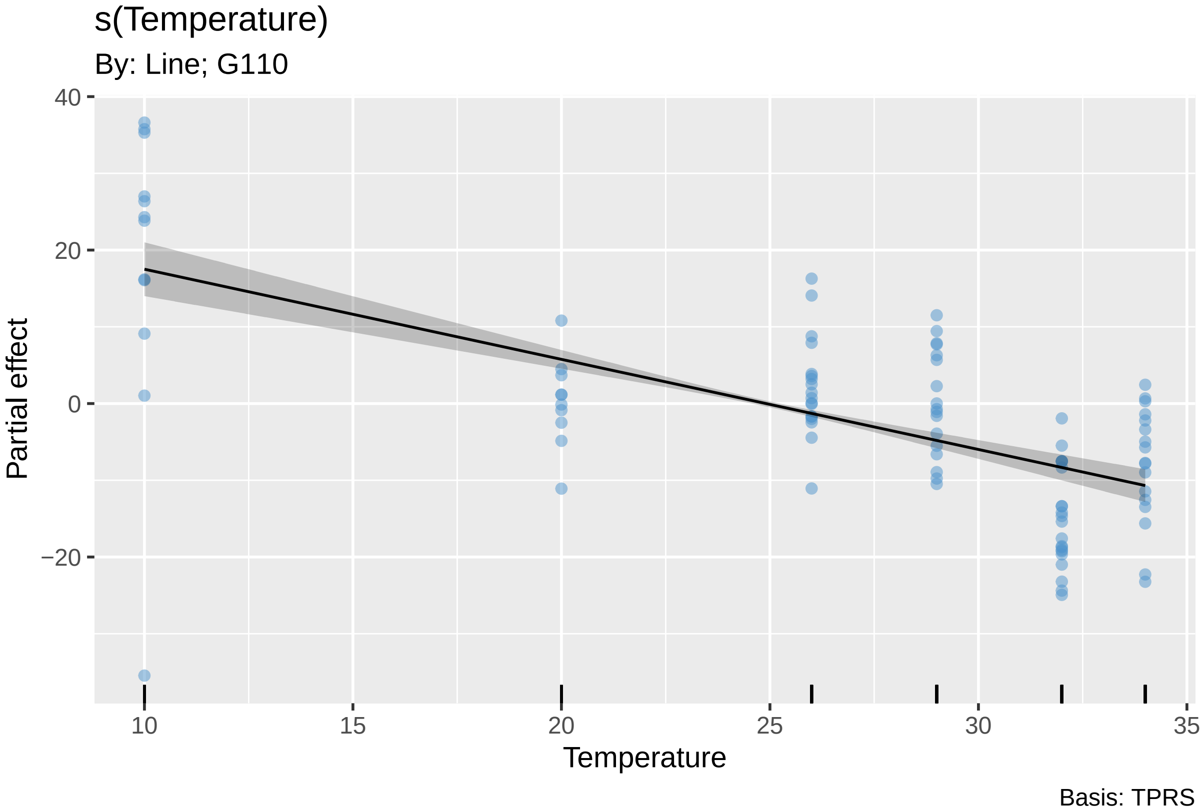
```r
for (i in 1:length(smooths(gam_width))) {
  out <- draw(gam_width, select = i, residuals = TRUE)
  print(out) }
```

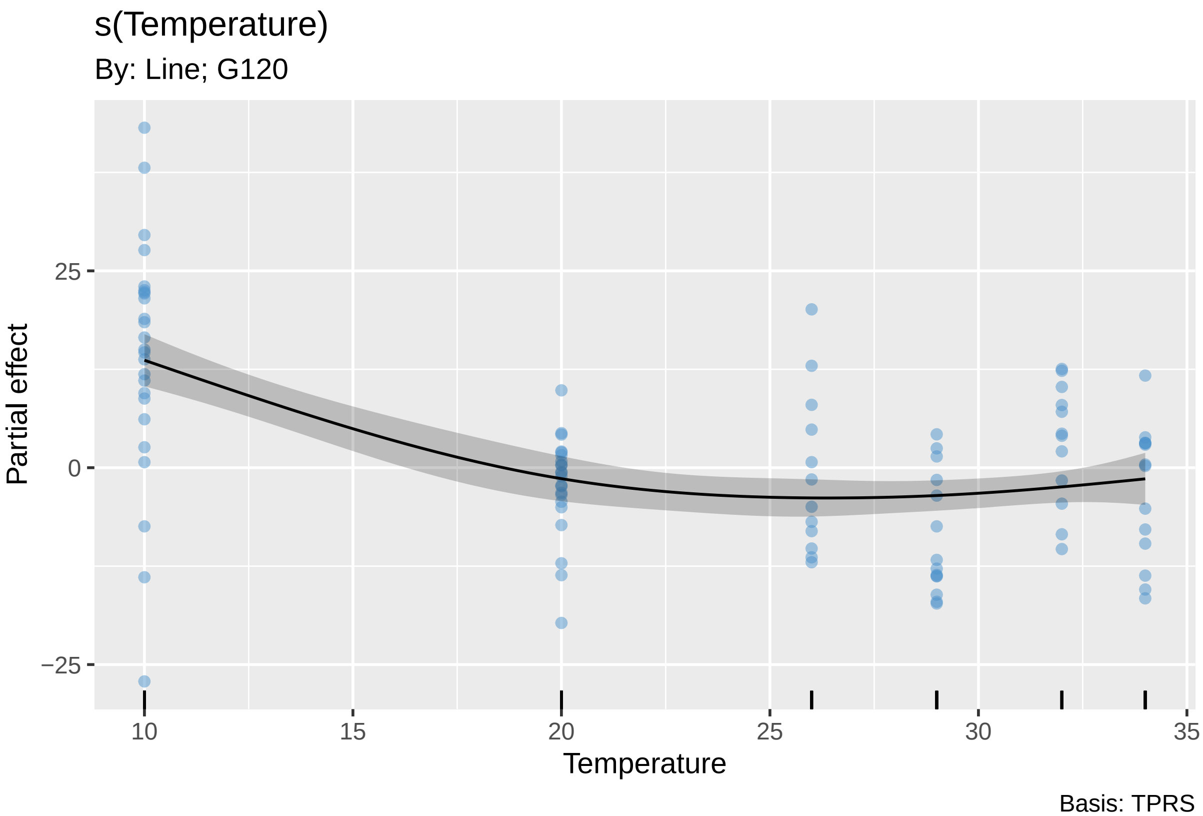


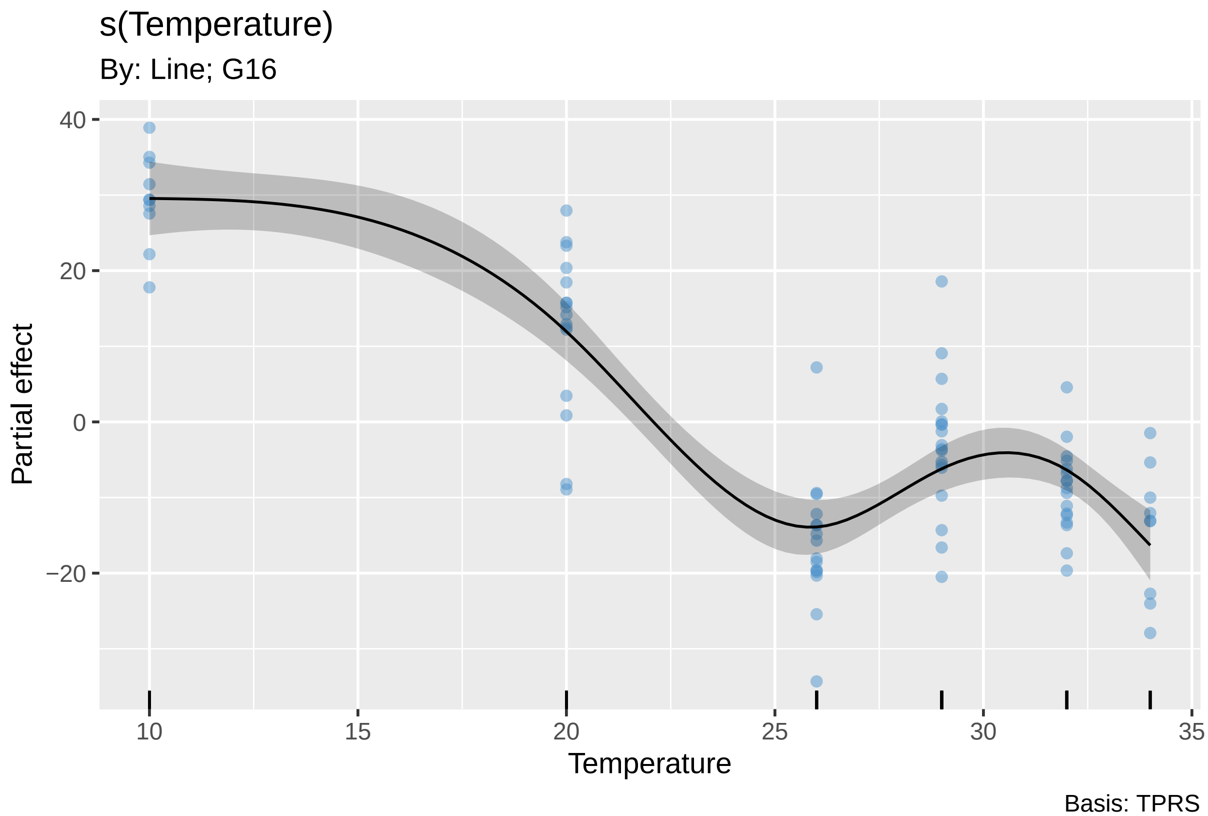
s(Temperature)
By: Line; G102
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

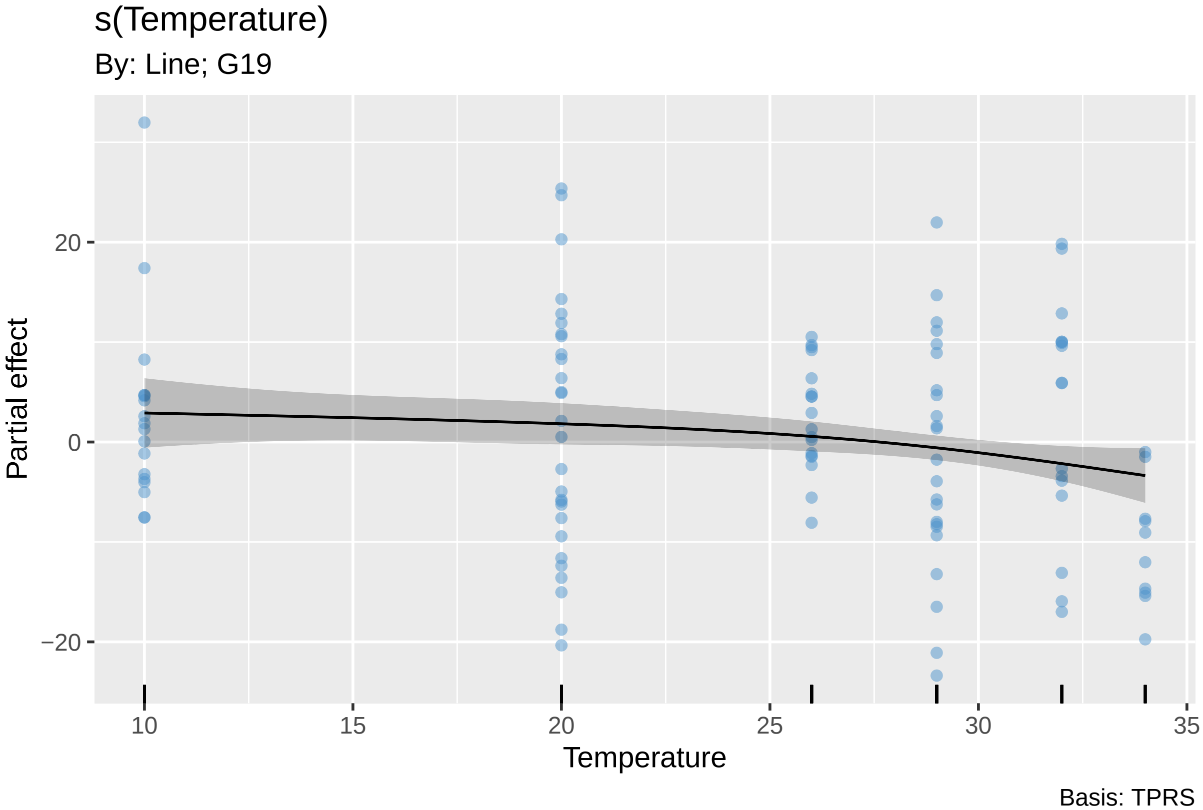


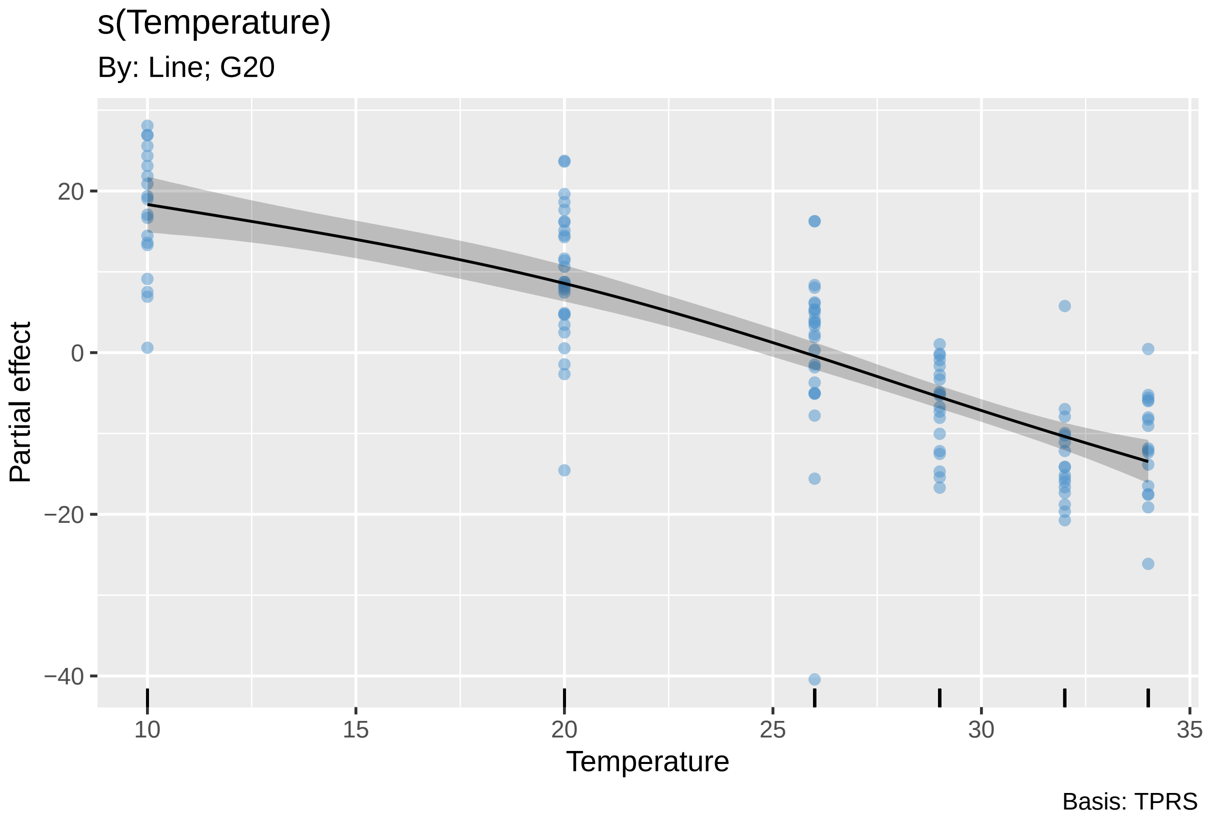
s(Temperature)
By: Line; G104
Partial effect
Temperature
40
20
0
−20
10
15
20
25
30
35
Basis: TPRS

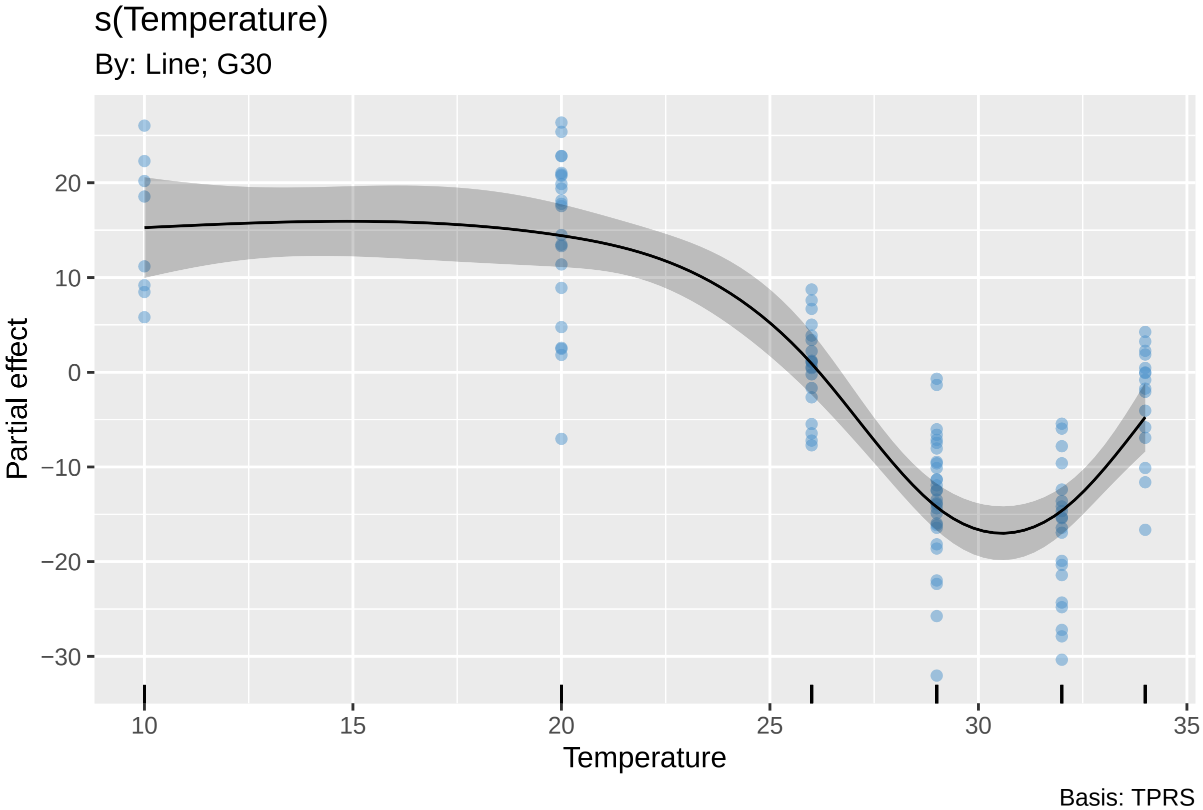

s(Temperature)
By: Line; G110
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

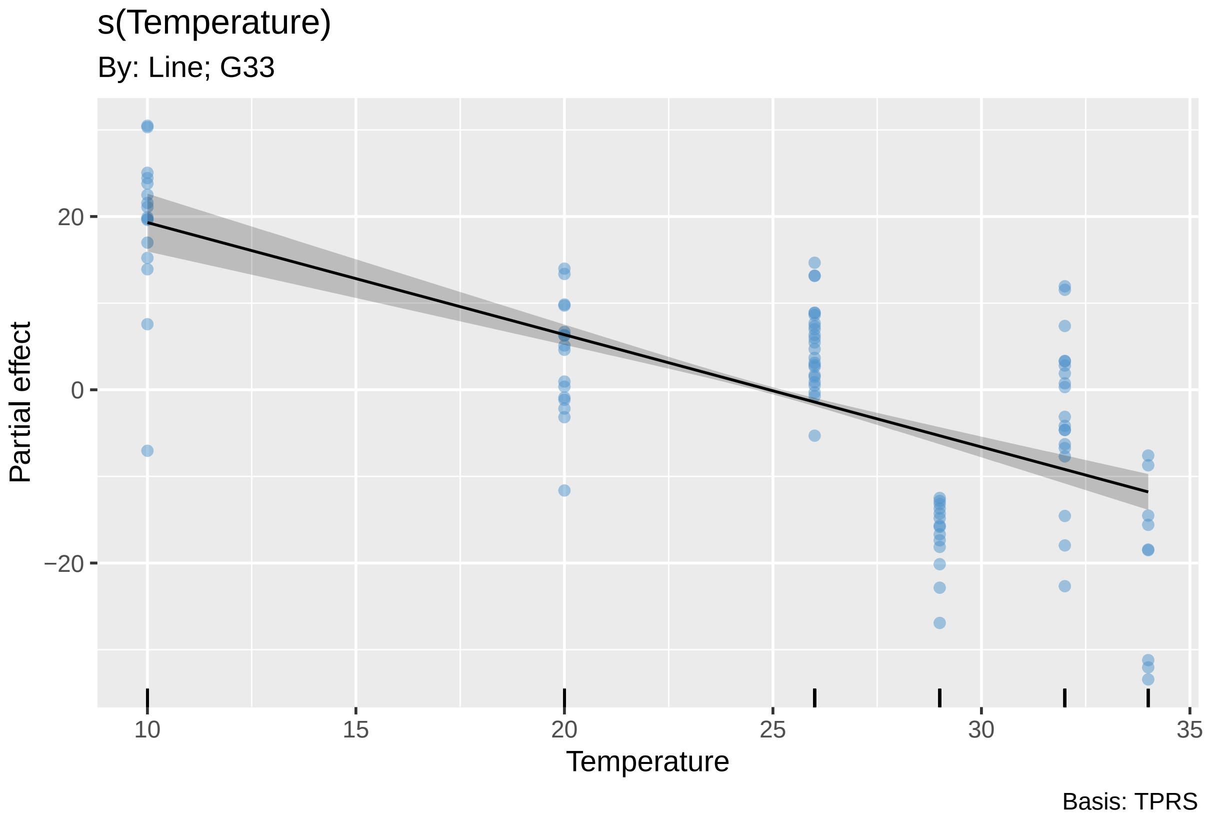

s(Temperature)
By: Line; G120
Partial effect
25
0
−25
10
15
20
25
30
35
Temperature
Basis: TPRS

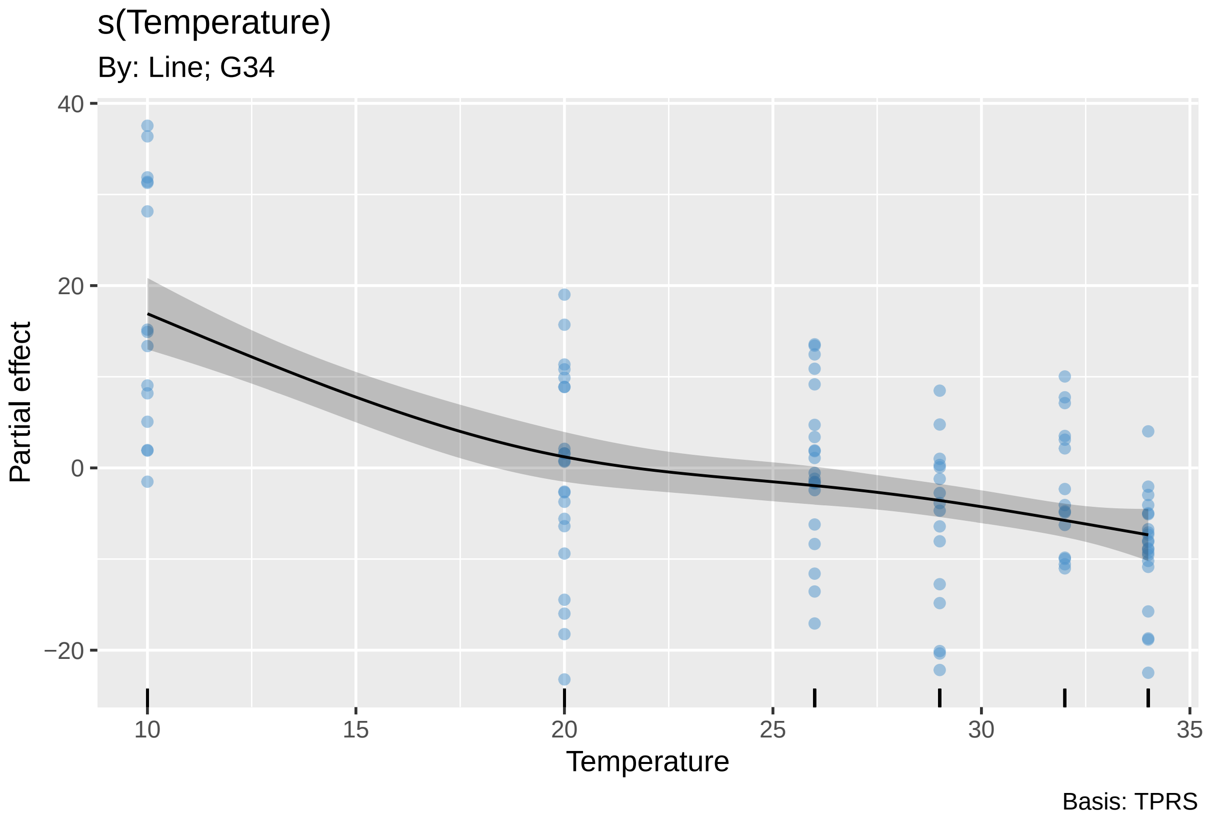

s(Temperature)
By: Line; G16
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G19
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

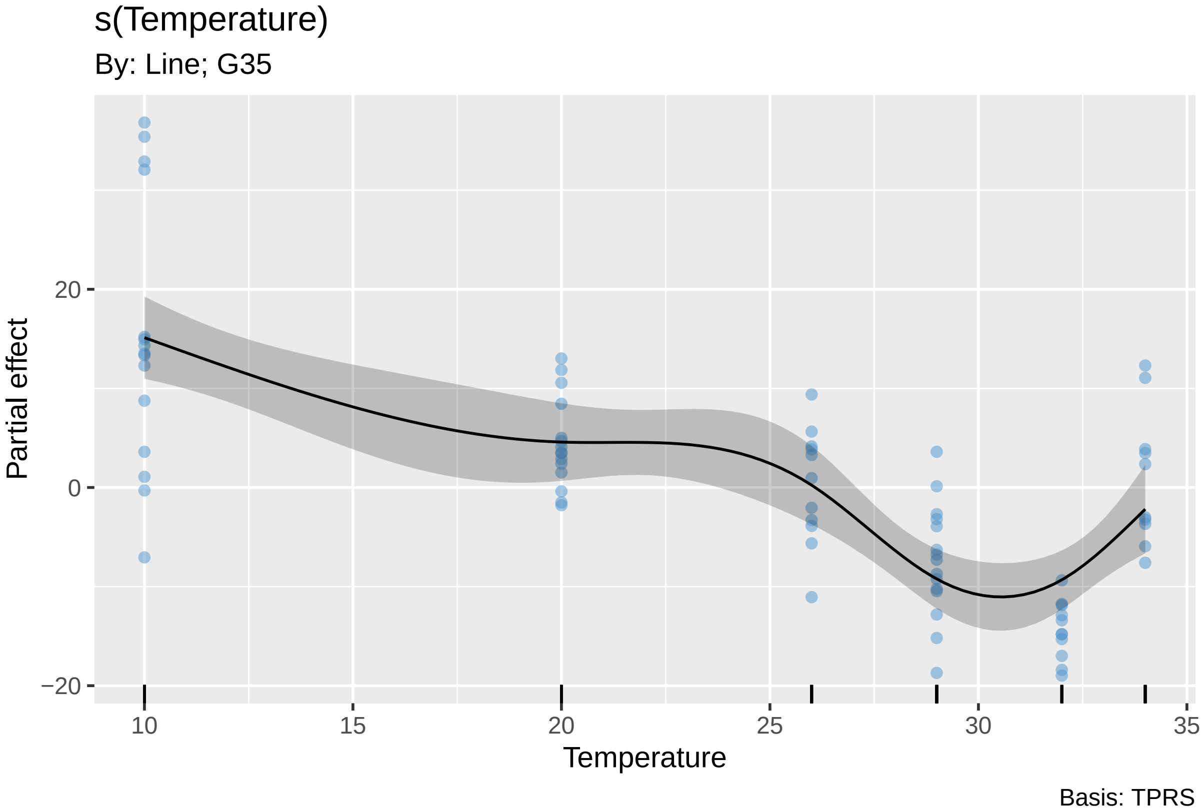

s(Temperature)
By: Line; G20
Partial effect
20
0
−20
−40
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G30
Partial effect
20
10
0
−10
−20
−30
10
15
20
25
30
35
Temperature
Basis: TPRS

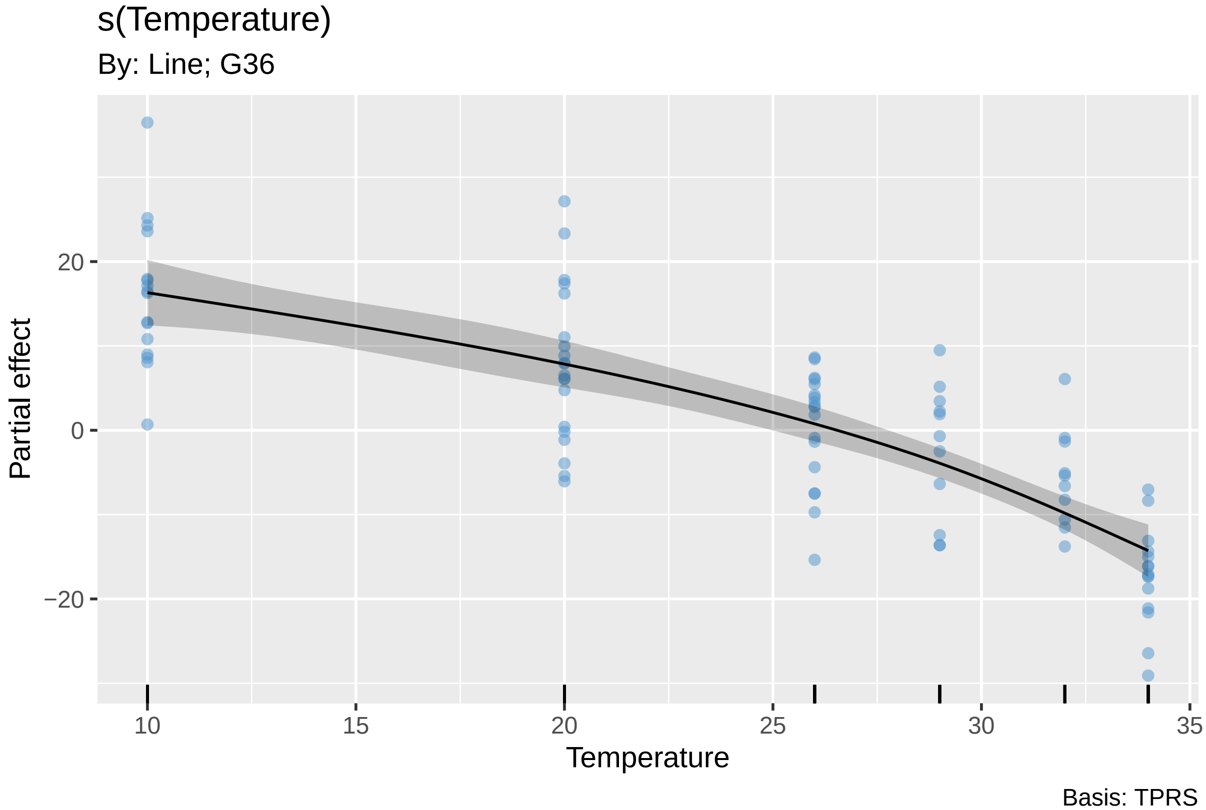

s(Temperature)
By: Line; G33
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

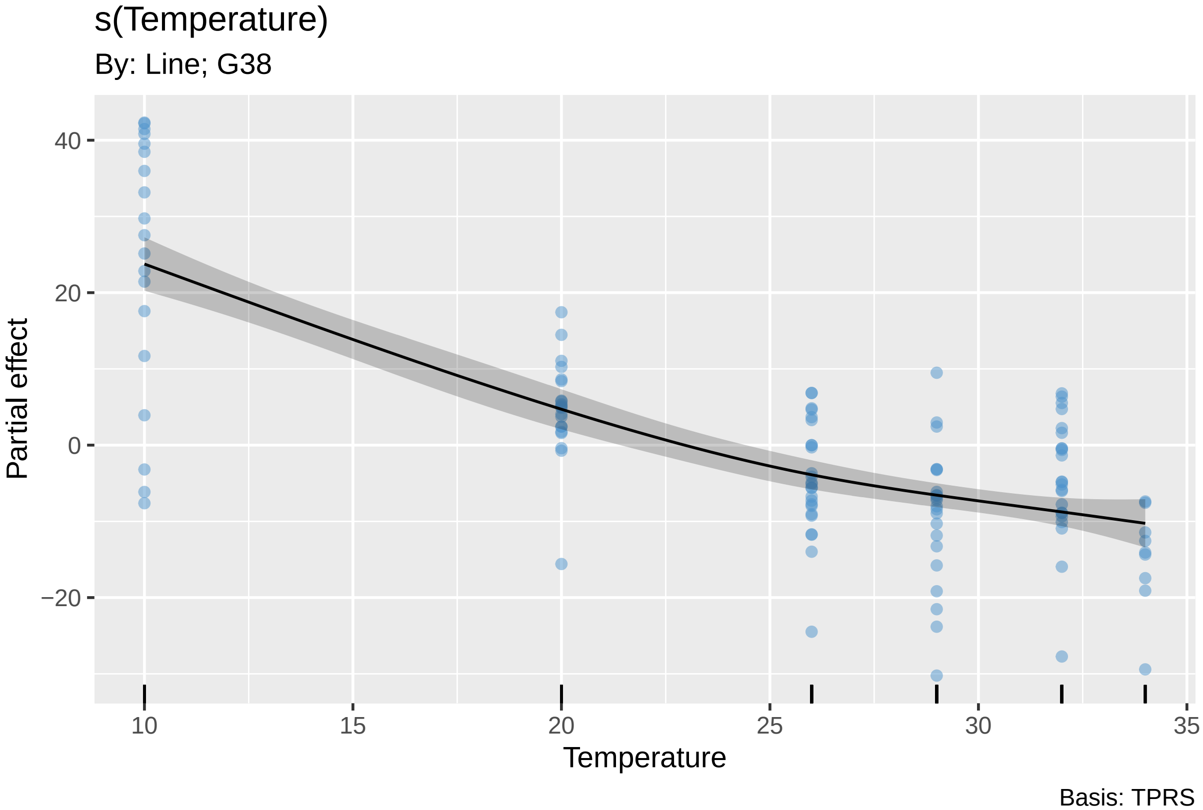


s(Temperature)
By: Line; G34
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

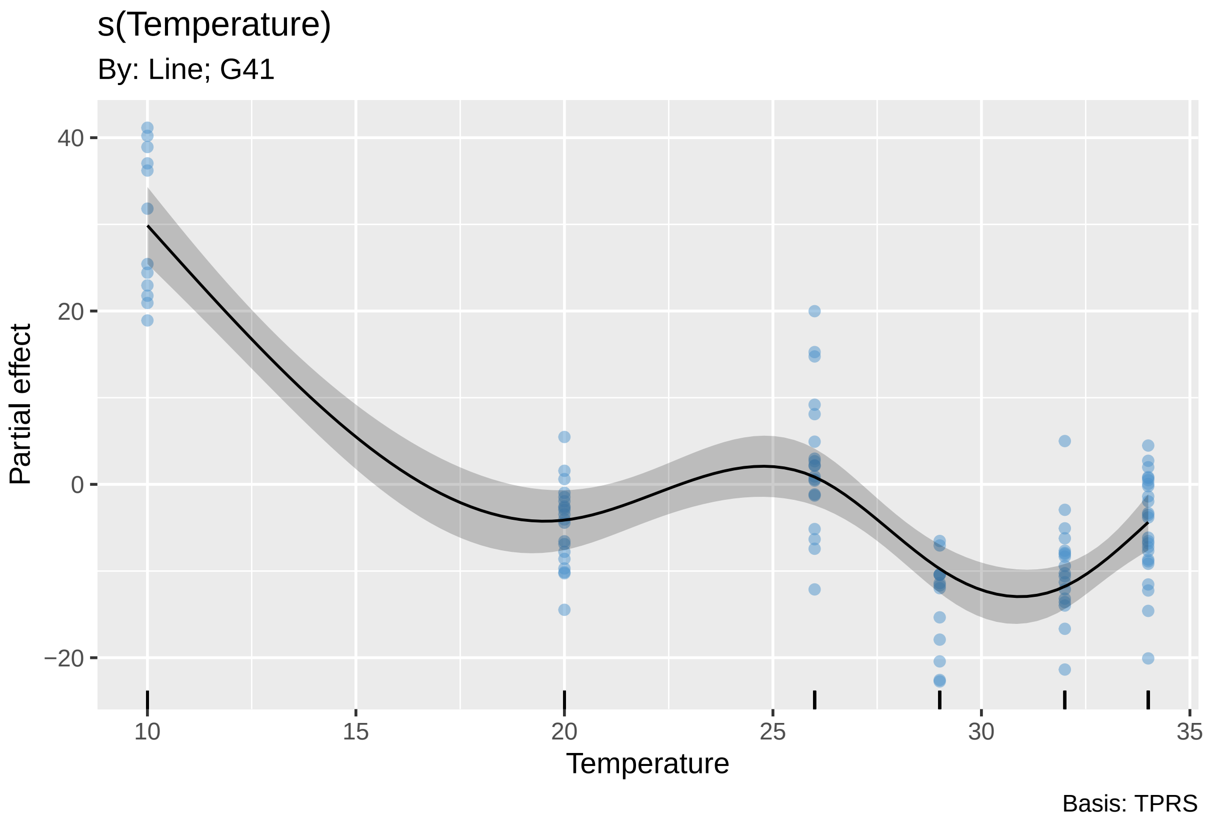


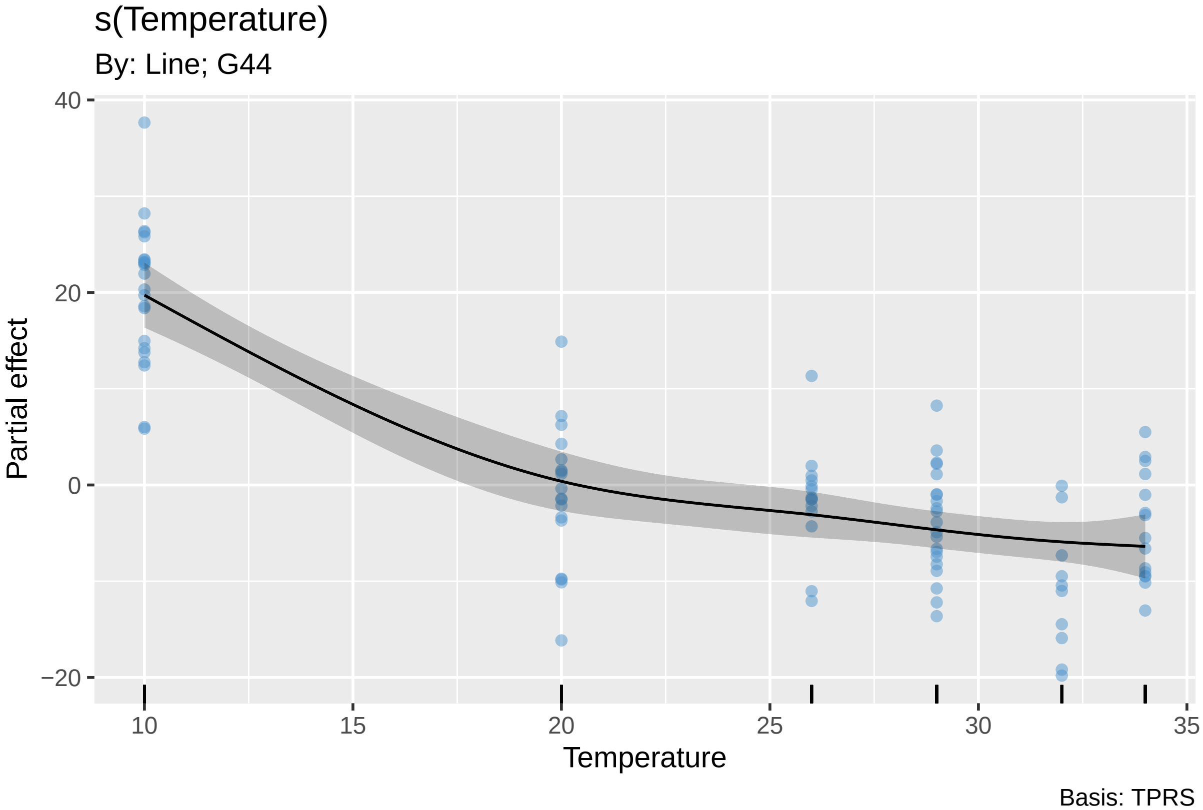
s(Temperature)
By: Line; G35
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

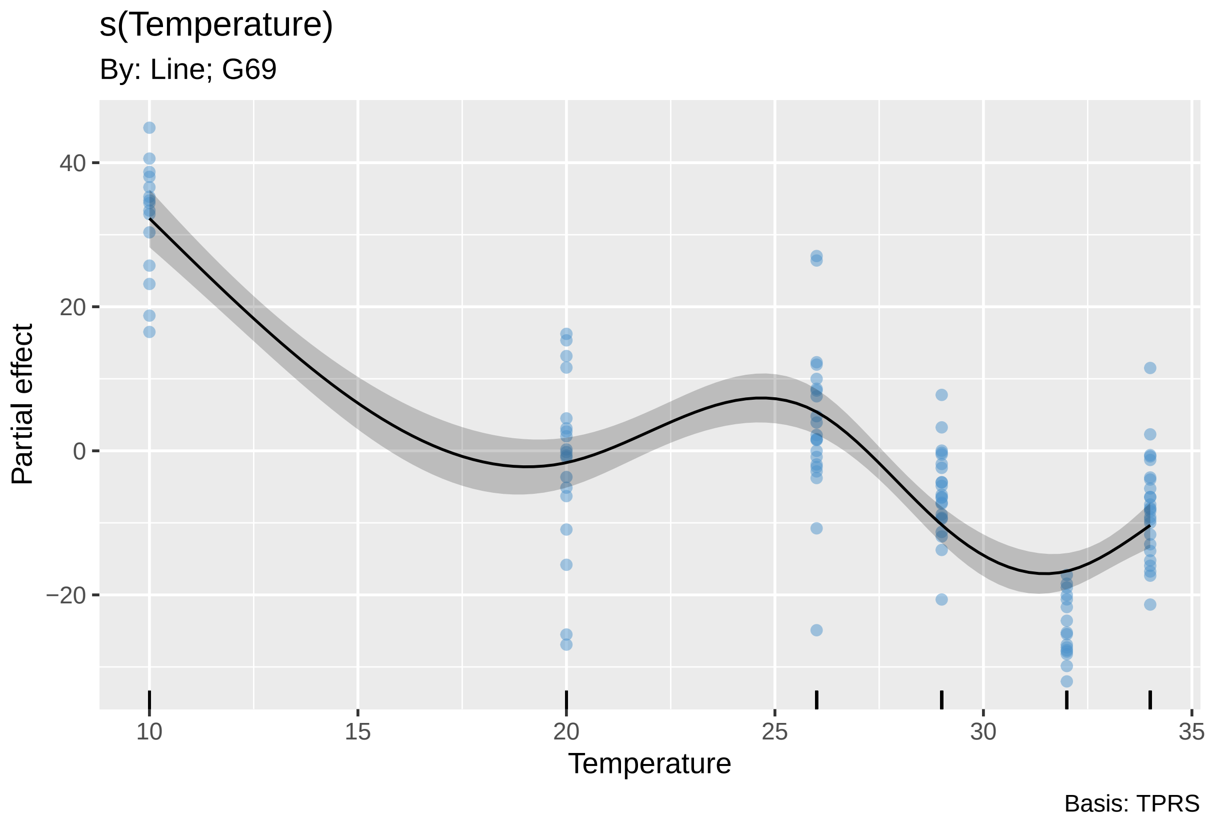

s(Temperature)
By: Line; G36
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS



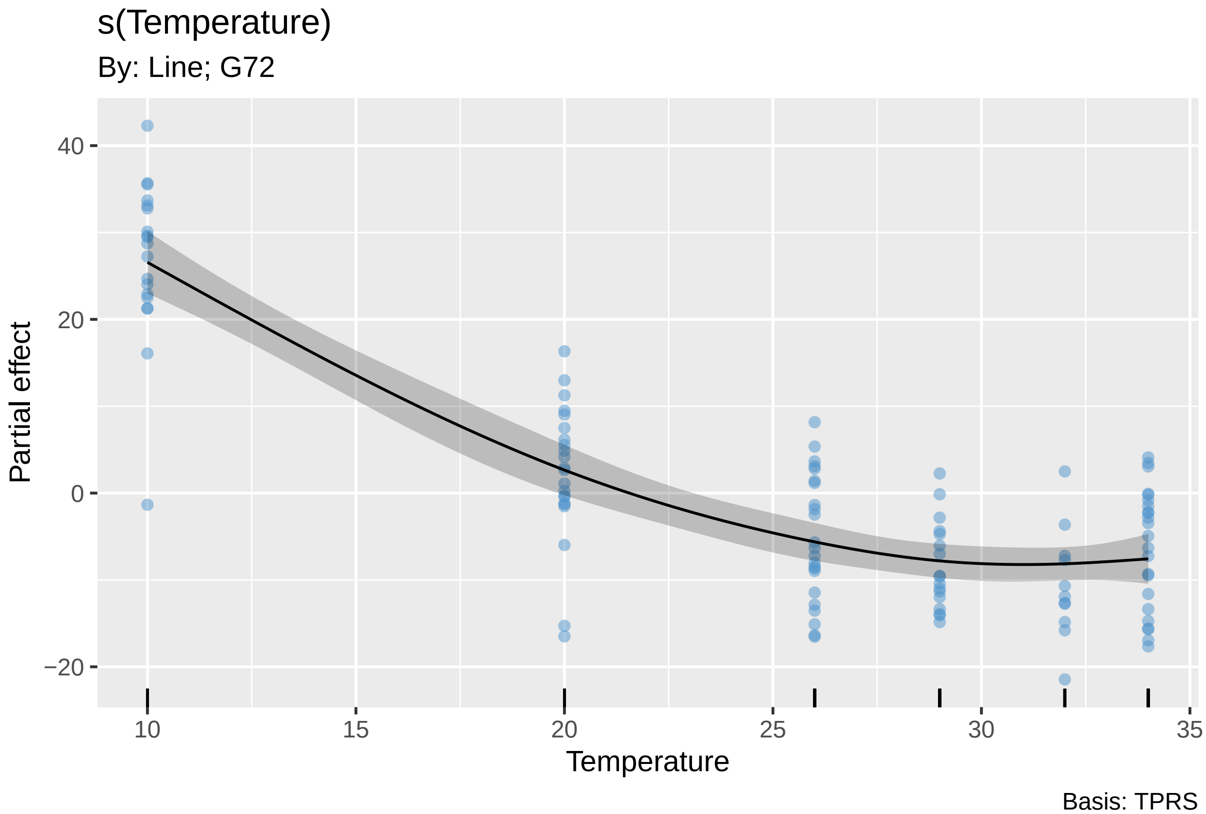
s(Temperature)
By: Line; G38
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

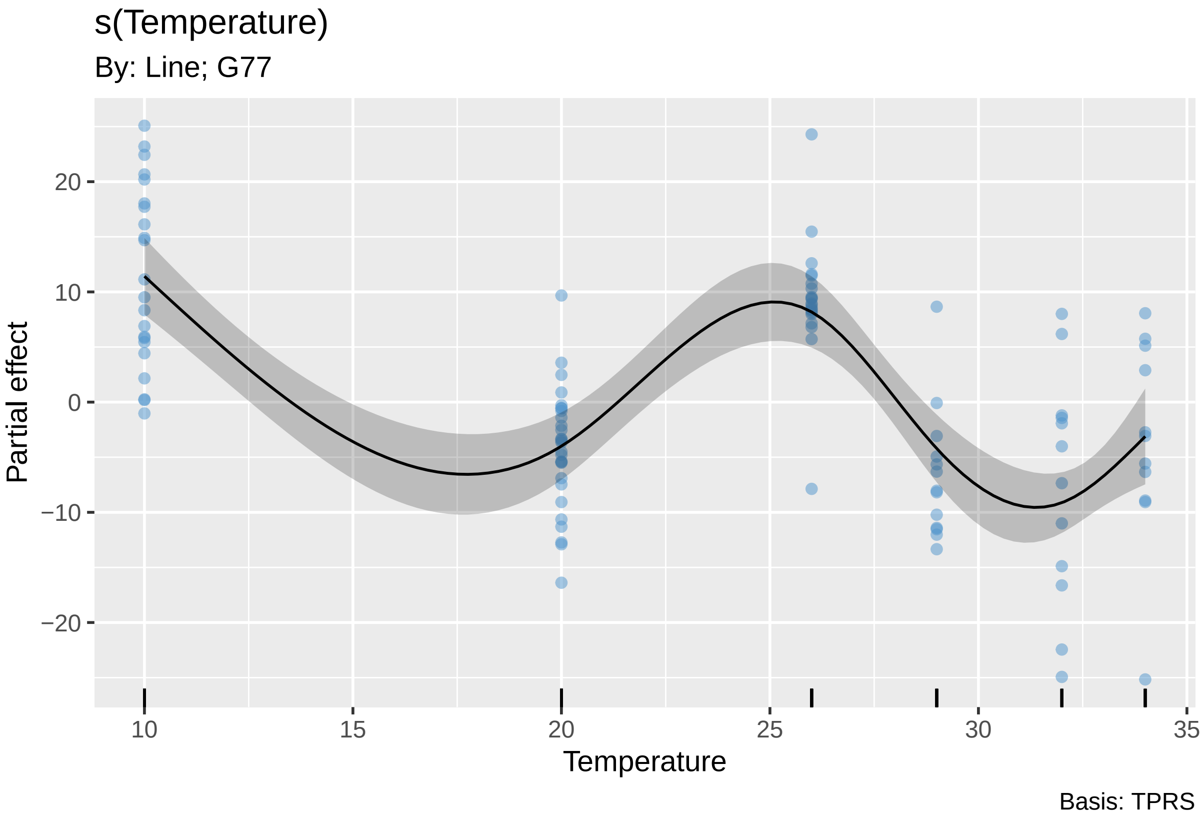

s(Temperature)
By: Line; G41
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

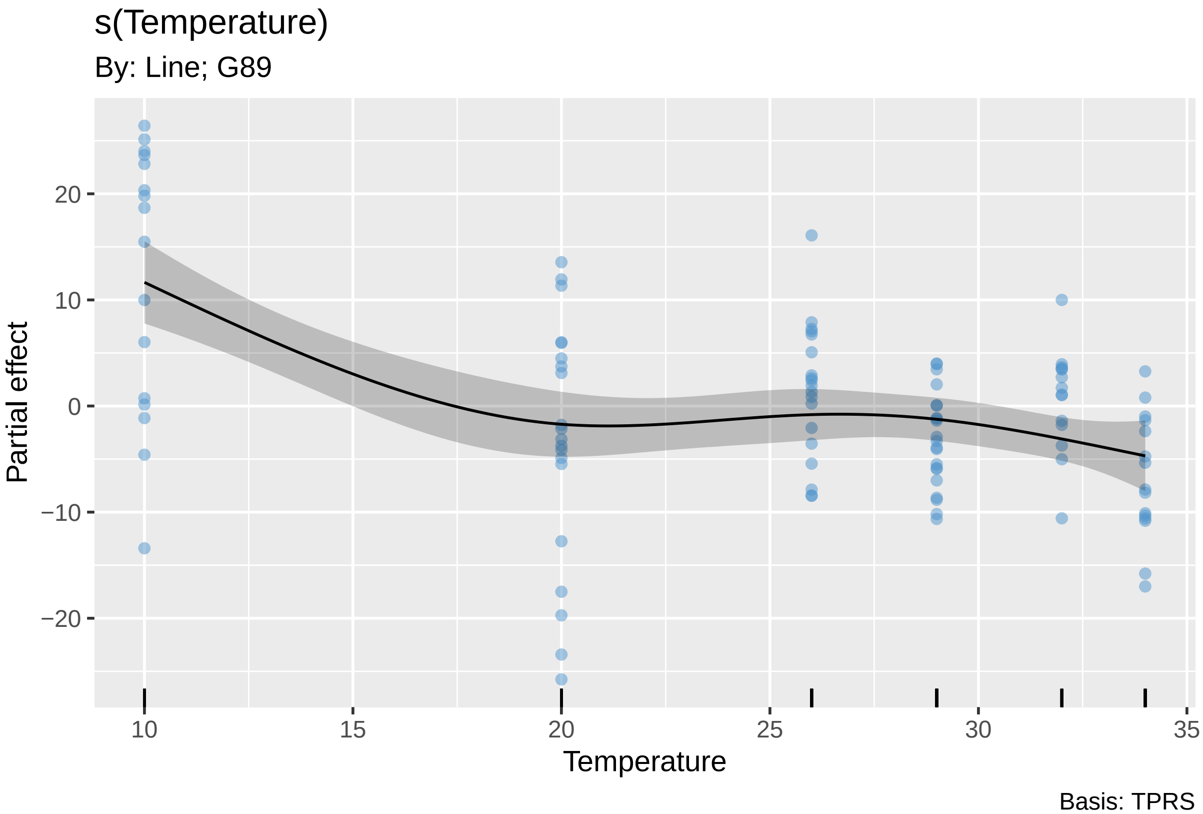

s(Temperature)
By: Line; G44
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

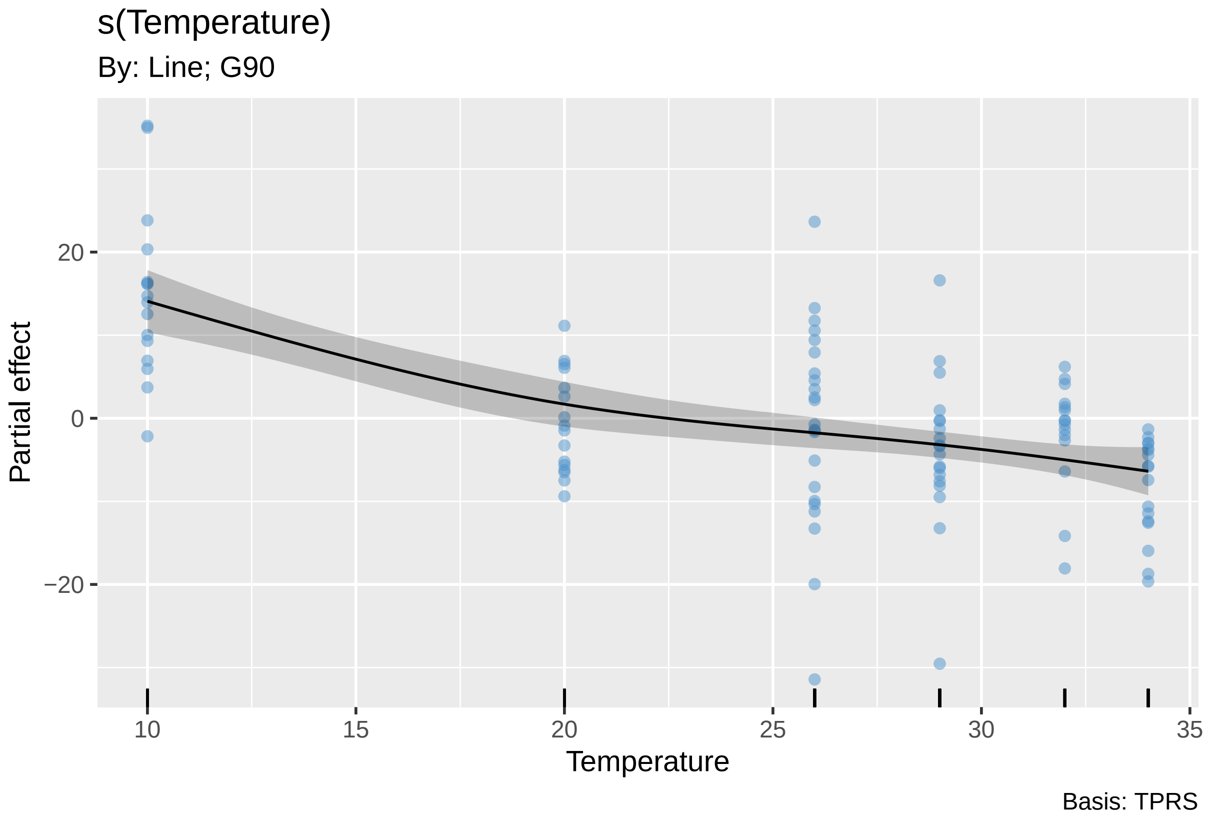


s(Temperature)
By: Line; G69
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G72
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G77
Partial effect
20
10
0
−10
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G89
Partial effect
20
10
0
−10
−20
10
15
20
25
30
35
Temperature
Basis: TPRS



Again, we will assess whether the inclusion of the interaction terms improves the model's ability to predict the data.

```
gam_width_noint <- gam(formula = mean_minor ~ Line + s(Temperature, k = 5, bs = 'tp'), data = morph_data

AIC(gam_width, gam_width_noint)
```

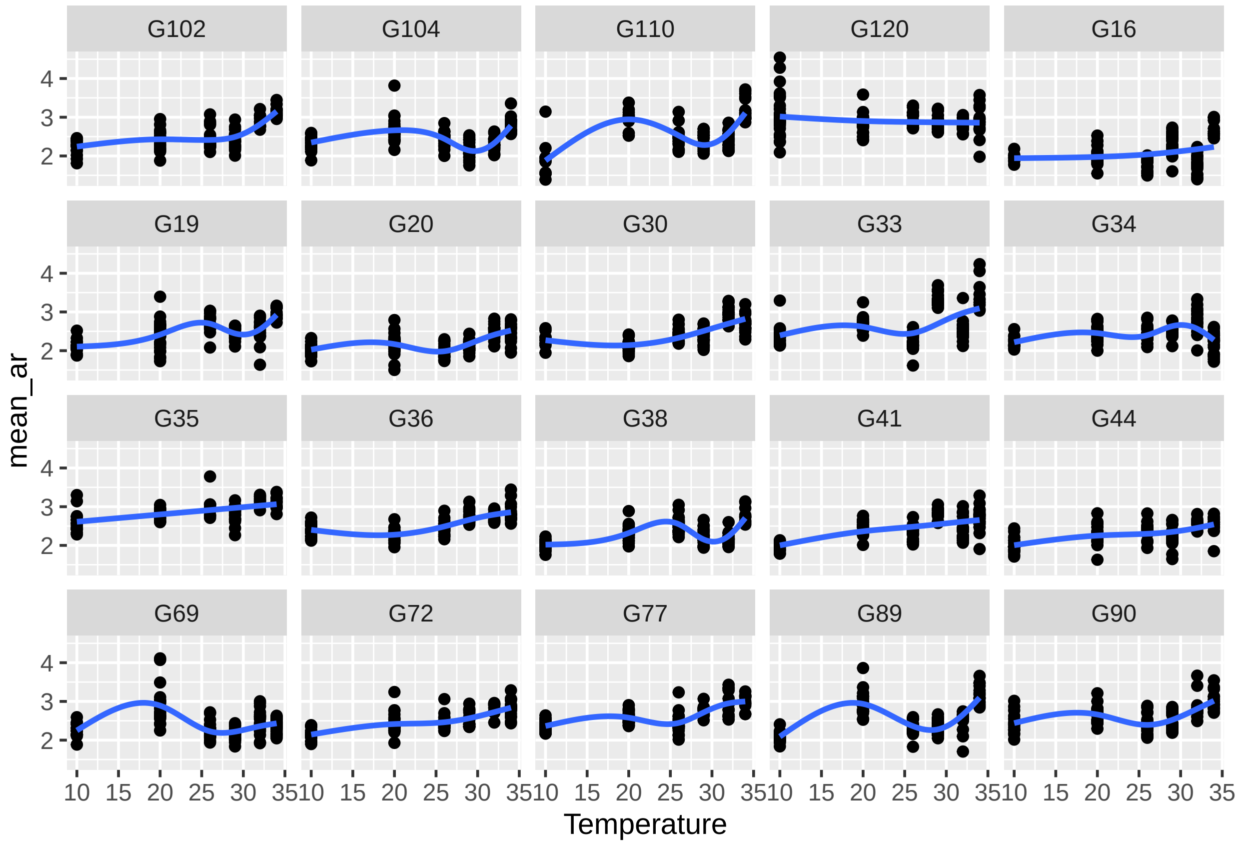
```
##                        df      AIC
## gam_width       77.27334 15133.30
## gam_width_noint 24.99329 15598.34
```

And, again we see that AIC score is much lower for the model including the interaction (ΔAIC = 465.0388242).

Aspect Ratio

Now we will examine how the aspect ratio of the paramecium changes with temperature.

```
ggplot(data = morph_data, aes(x = Temperature, y = mean_ar)) + geom_point() + geom_smooth(method = 'gam
  facet_wrap(facets = "Line", nrow = 4, ncol = 5)
```



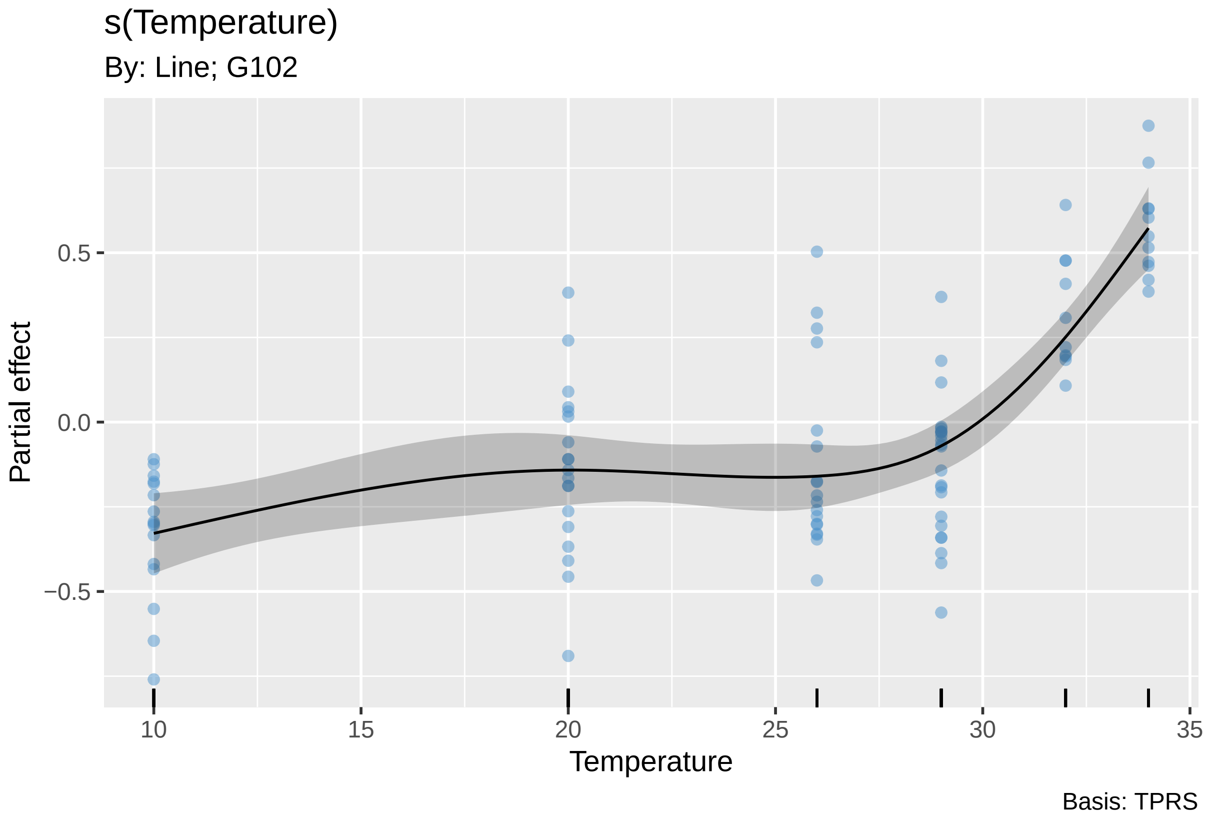
Overall, across the different lines, the general trend is for an overall increase in aspect ratio with temperature. Fitting a GAM gives:

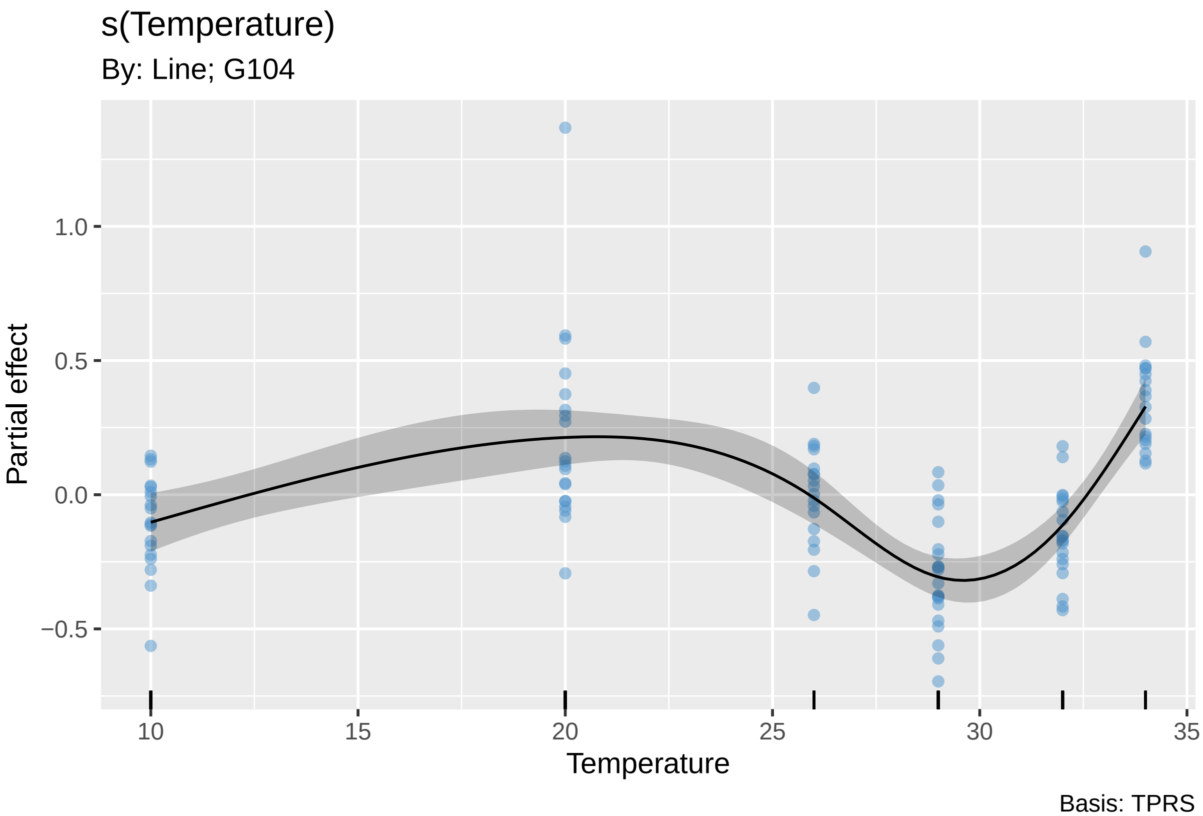
```
gam_ar <- gam(formula = mean_ar ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = morph_data

summary(gam_ar)
```

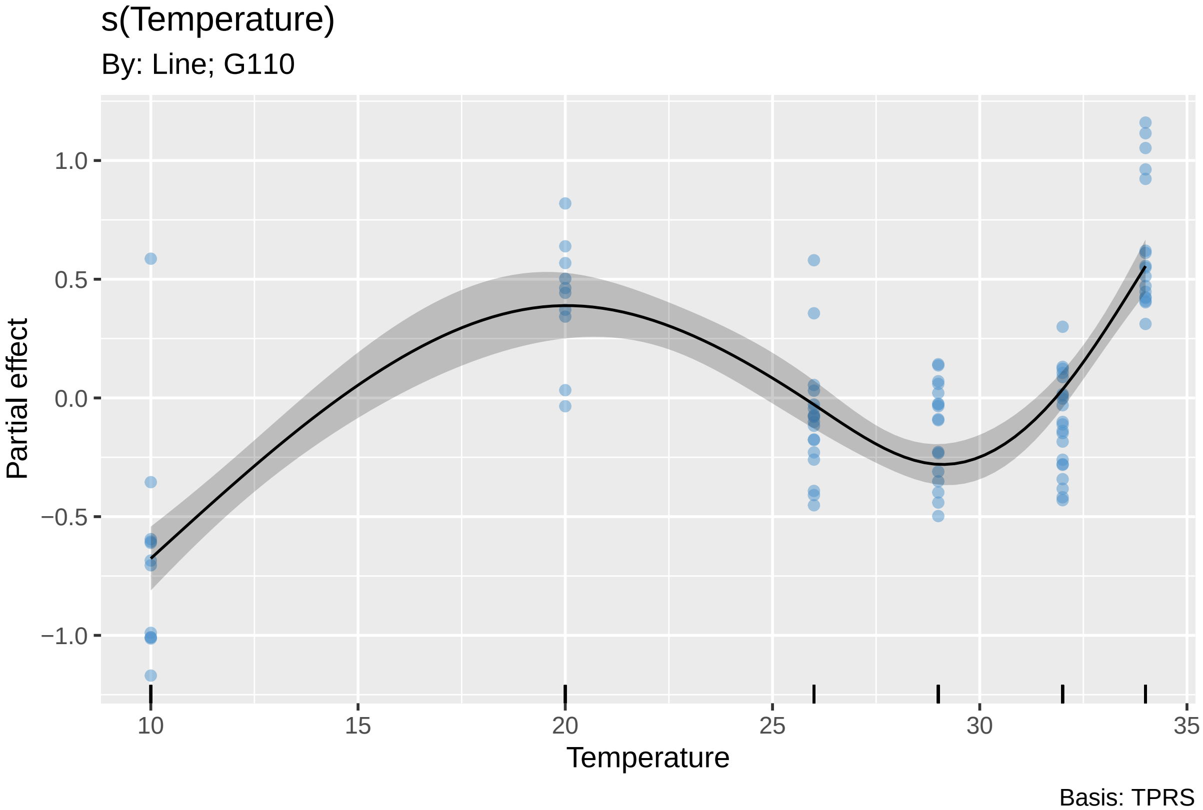
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean_ar ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.570162   0.026828  95.801  < 2e-16 ***
## LineG104     -0.120570   0.036183  -3.332 0.000877 ***
## LineG110     -0.012073   0.038846  -0.311 0.755982
## LineG120      0.327496   0.038102   8.595  < 2e-16 ***
## LineG16      -0.496818   0.039108 -12.704  < 2e-16 ***
## LineG19      -0.053156   0.036817  -1.444 0.148946
## LineG20      -0.368805   0.035638 -10.349  < 2e-16 ***
## LineG30      -0.127486   0.037273  -3.420 0.000638 ***
## LineG33       0.119123   0.037808   3.151 0.001652 **
## LineG34      -0.141623   0.036761  -3.853 0.000121 ***
## LineG35       0.322287   0.040136   8.030 1.64e-15 ***
```

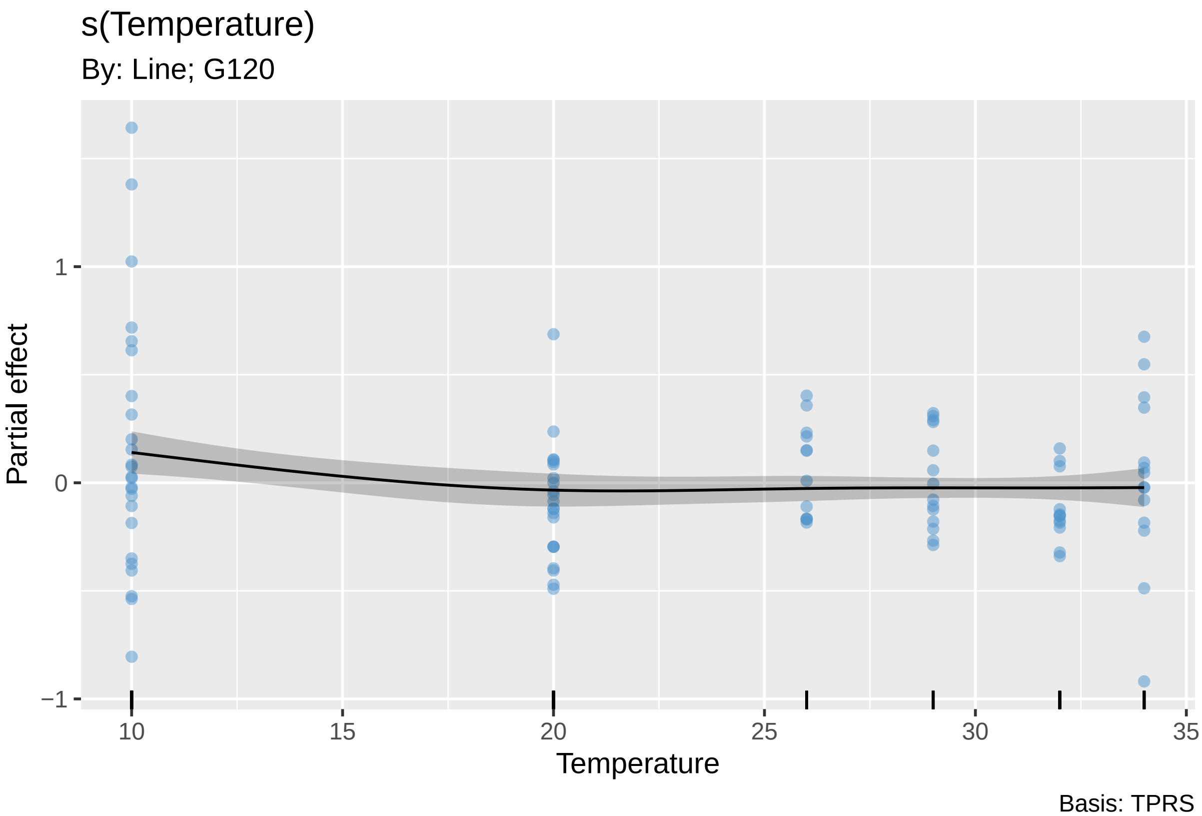
```
## LineG36     -0.007736   0.038614  -0.200 0.841242
## LineG38     -0.240060   0.036317  -6.610 4.90e-11 ***
## LineG41     -0.130129   0.037200  -3.498 0.000479 ***
## LineG44     -0.259106   0.037517  -6.906 6.64e-12 ***
## LineG69     -0.164042   0.036184  -4.533 6.14e-06 ***
## LineG72     -0.054556   0.036292  -1.503 0.132931
## LineG77      0.089908   0.037489   2.398 0.016564 *
## LineG89     -0.005389   0.037007  -0.146 0.884240
## LineG90      0.056583   0.037043   1.528 0.126793
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                             edf Ref.df      F  p-value
## s(Temperature):LineG102 3.487  3.816 27.215  < 2e-16 ***
## s(Temperature):LineG104 3.884  3.990 19.201  < 2e-16 ***
## s(Temperature):LineG110 3.892  3.992 49.412  < 2e-16 ***
## s(Temperature):LineG120 2.141  2.540  3.657  0.02997 *
## s(Temperature):LineG16  2.307  2.753  5.638  0.00285 **
## s(Temperature):LineG19  3.873  3.987 21.364  < 2e-16 ***
## s(Temperature):LineG20  3.523  3.838 15.296  < 2e-16 ***
## s(Temperature):LineG30  2.839  3.286 27.635  < 2e-16 ***
## s(Temperature):LineG33  3.776  3.964 24.325  < 2e-16 ***
## s(Temperature):LineG34  3.819  3.975  8.817 2.81e-06 ***
## s(Temperature):LineG35  1.000  1.000 27.732 2.74e-07 ***
## s(Temperature):LineG36  2.742  3.164 18.935  < 2e-16 ***
## s(Temperature):LineG38  3.926  3.996 18.274  < 2e-16 ***
## s(Temperature):LineG41  2.026  2.439 25.002  < 2e-16 ***
## s(Temperature):LineG44  1.000  1.000 45.011  < 2e-16 ***
## s(Temperature):LineG69  3.817  3.975 23.419  < 2e-16 ***
## s(Temperature):LineG72  2.643  3.072 27.777  < 2e-16 ***
## s(Temperature):LineG77  3.560  3.857 20.433  < 2e-16 ***
## s(Temperature):LineG89  3.857  3.984 48.478  < 2e-16 ***
## s(Temperature):LineG90  3.585  3.881 19.201  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.555   Deviance explained = 57.2%
## -REML = 324.18  Scale est. = 0.068125  n = 2100
```

```r
for (i in 1:length(smooths(gam_ar))) {
  out <- draw(gam_ar, select = i, residuals = TRUE)
  print(out) }
```

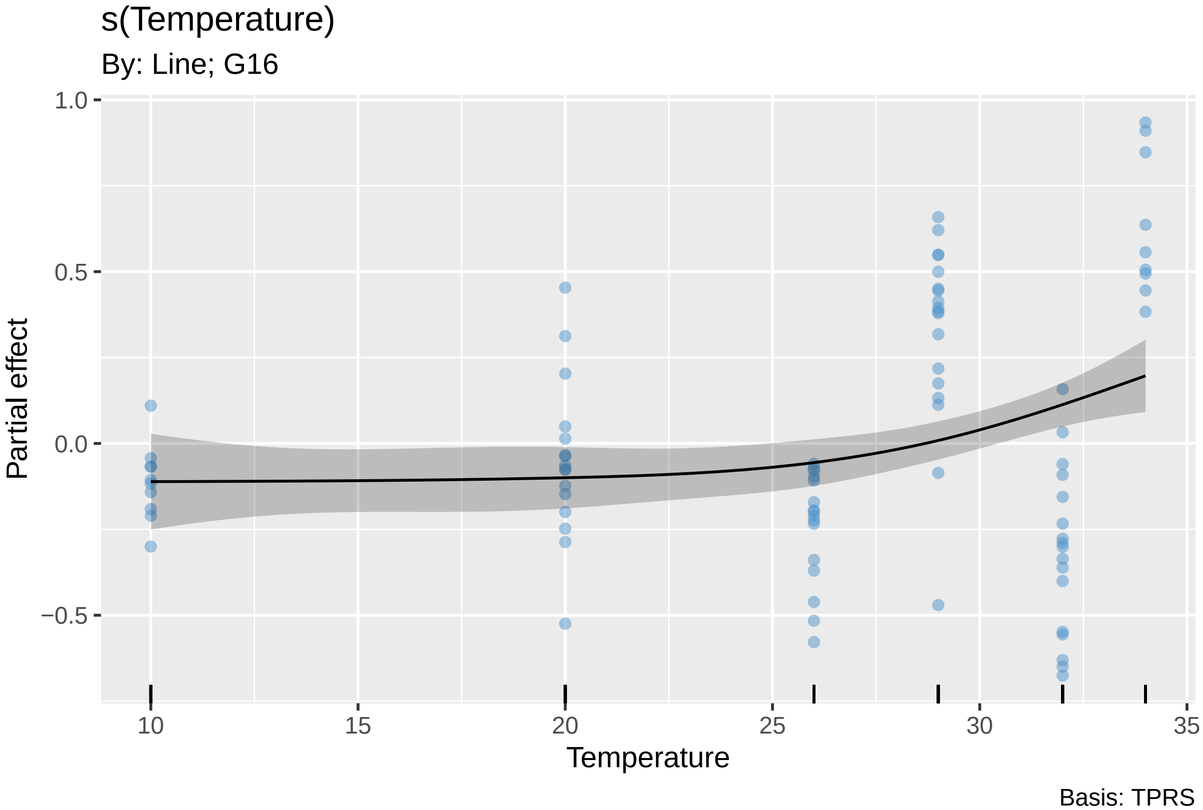

s(Temperature)
By: Line; G102
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

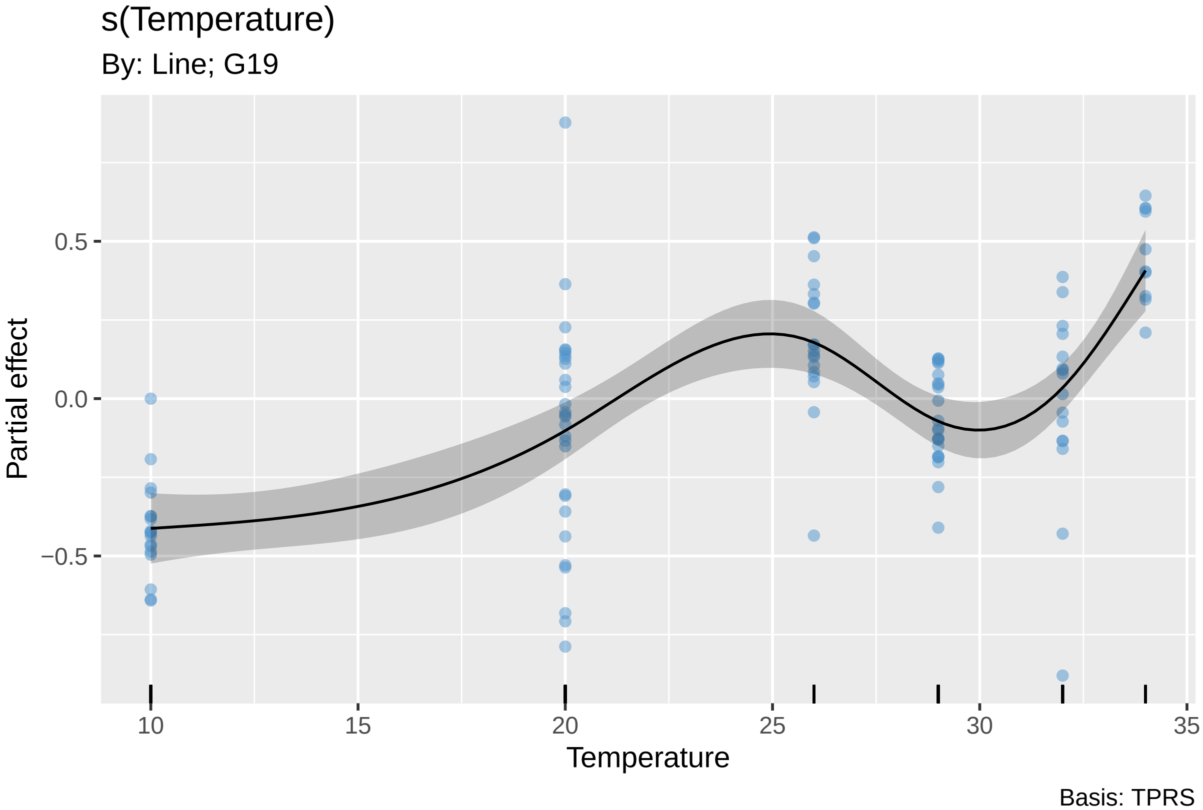

s(Temperature)
By: Line; G104
Partial effect
1.0
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

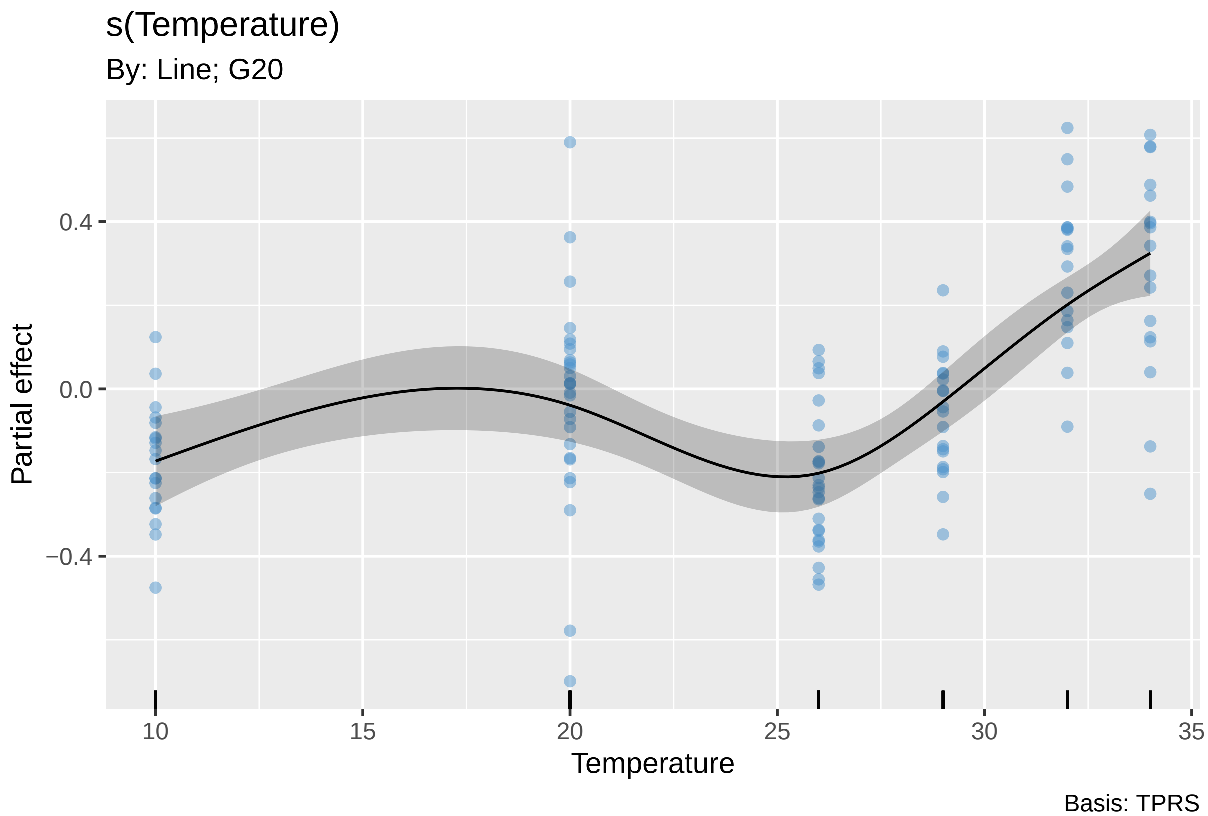

s(Temperature)
By: Line; G110
Partial effect
1.0
0.5
0.0
−0.5
−1.0
10
15
20
25
30
35
Temperature
Basis: TPRS

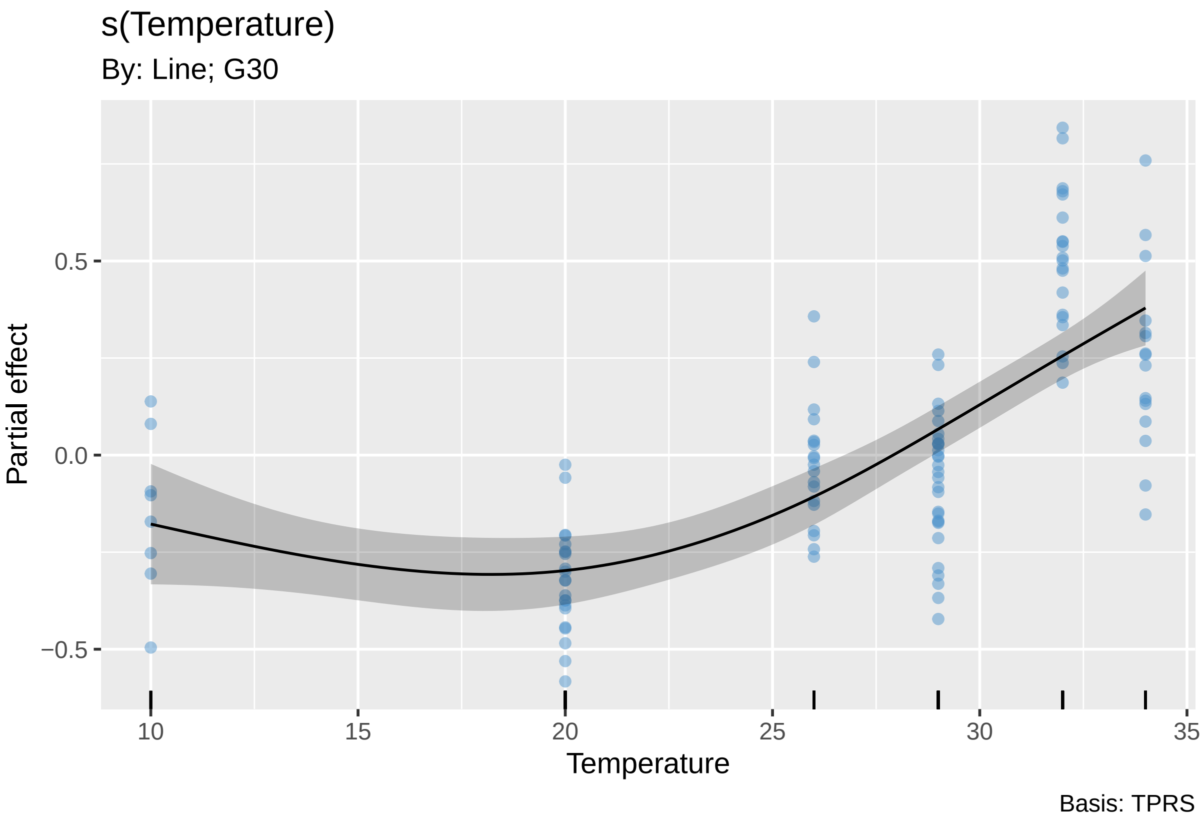


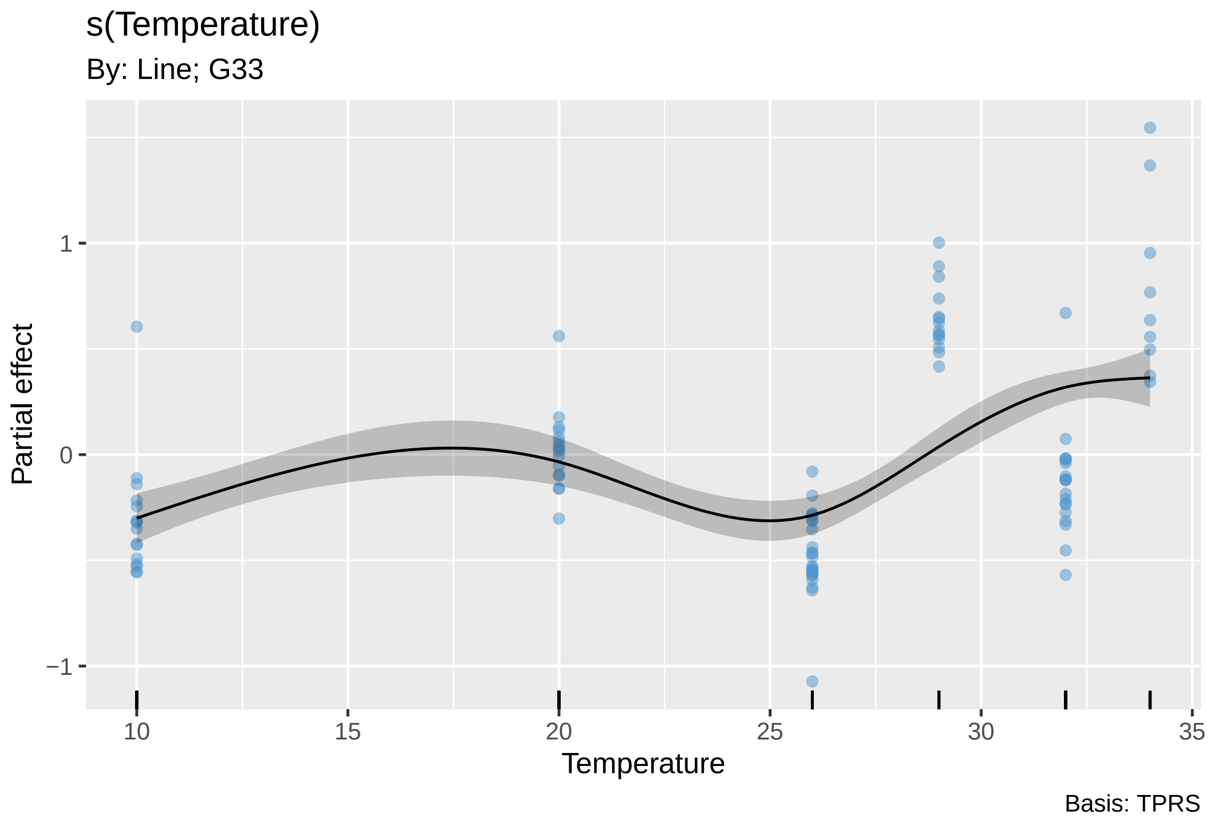
s(Temperature)
By: Line; G120
Partial effect
1
0
−1
10
15
20
25
30
35
Temperature
Basis: TPRS

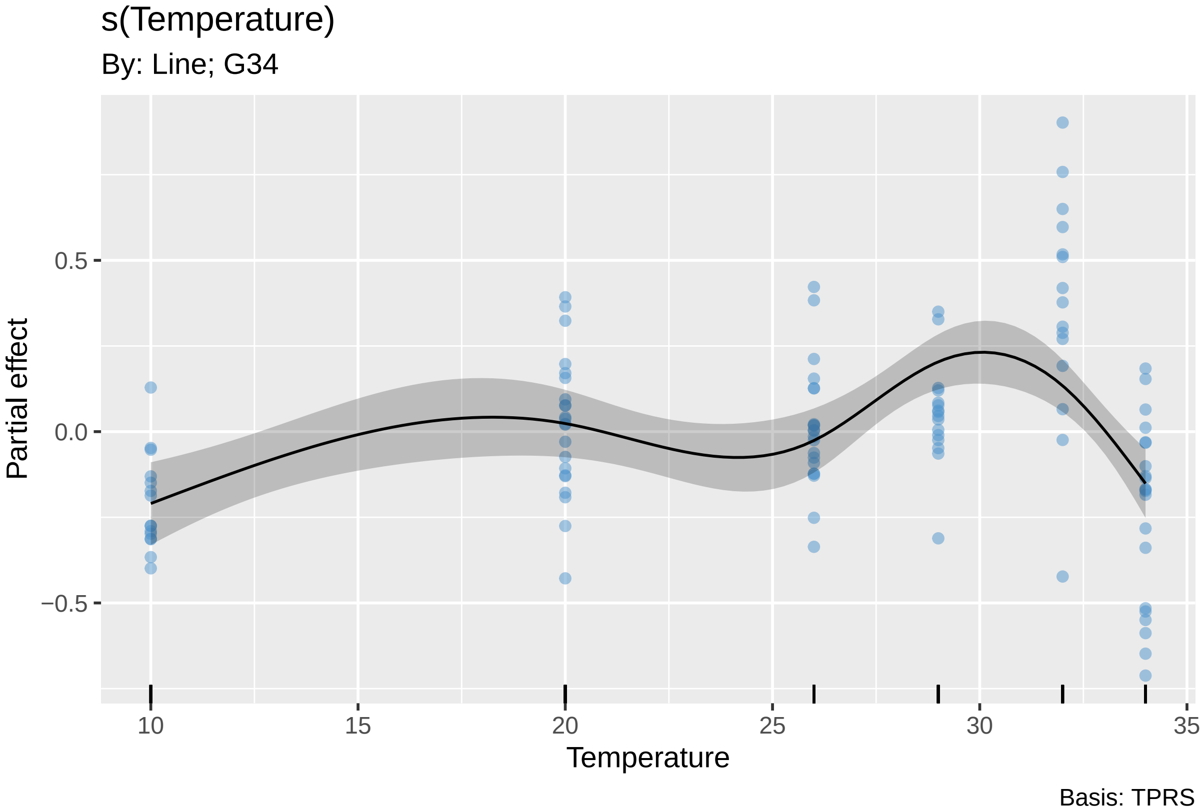

s(Temperature)
By: Line; G16
1.0
0.5
0.0
−0.5
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

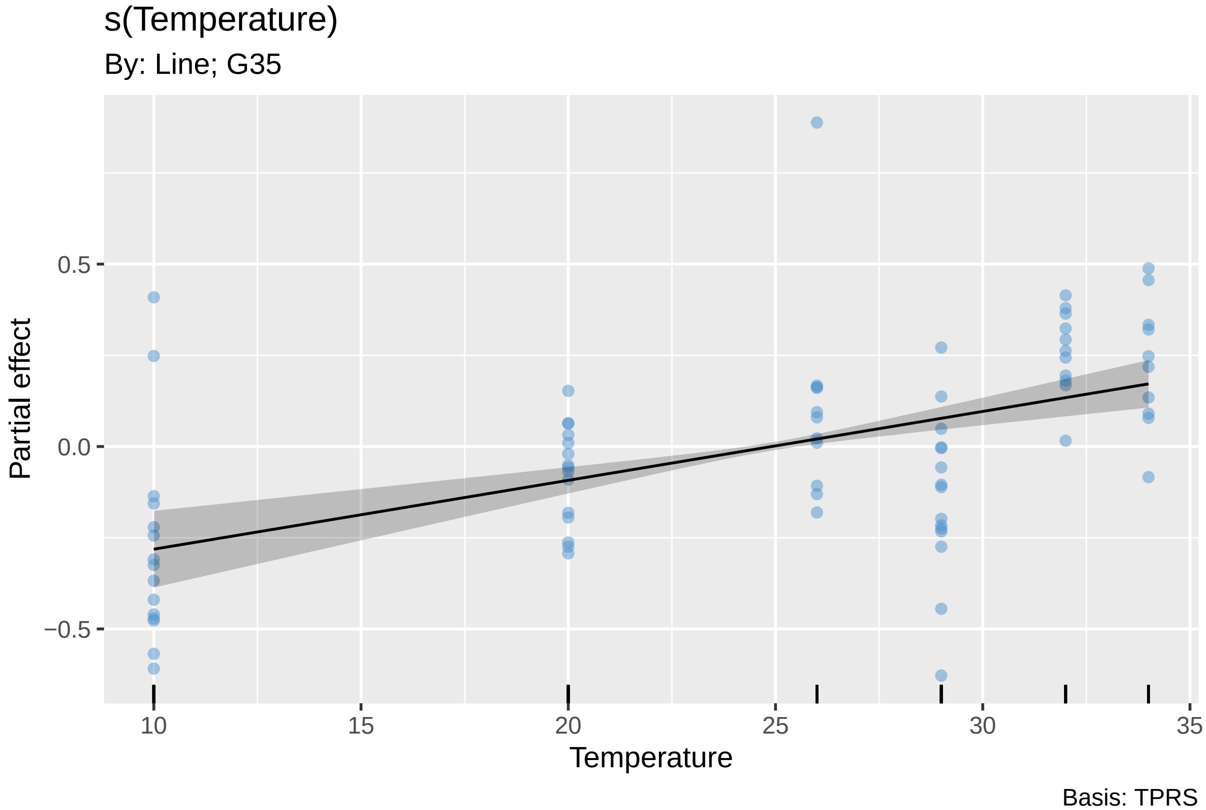

s(Temperature)
By: Line; G19
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

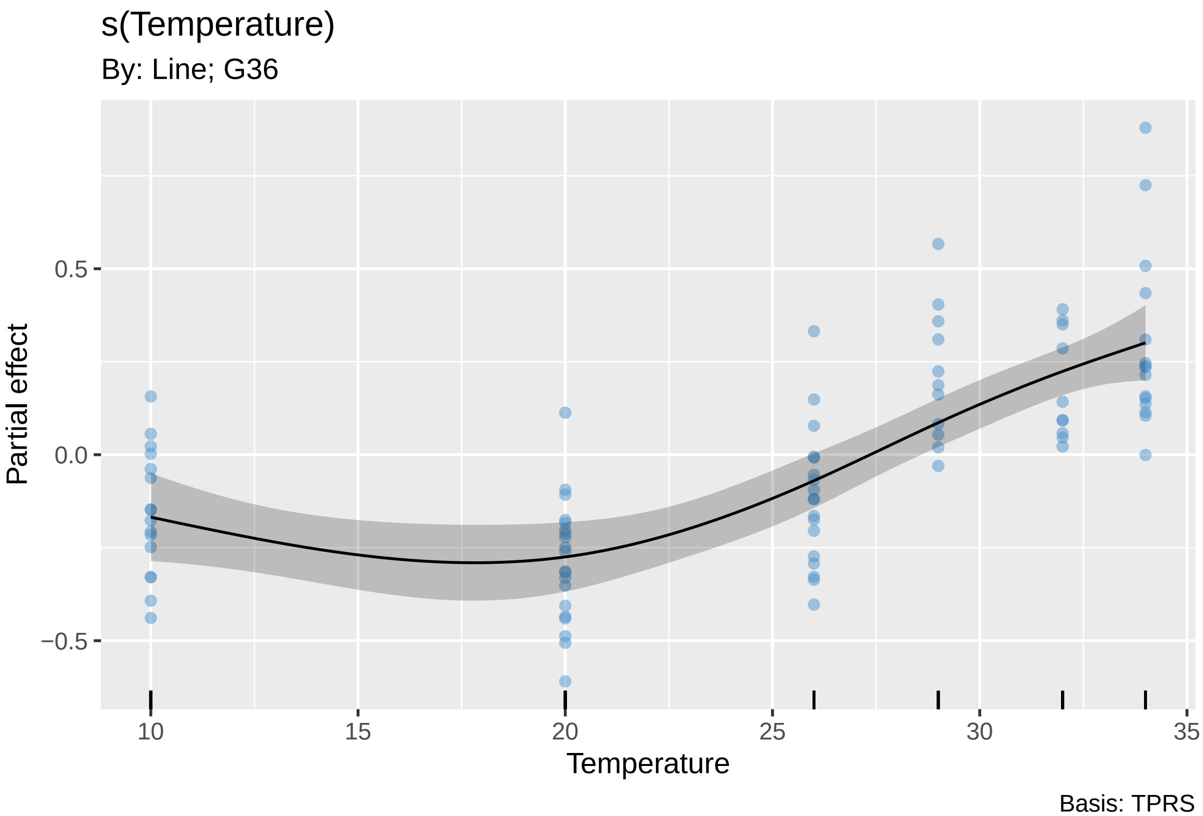

s(Temperature)
By: Line; G20
Partial effect
0.4
0.0
−0.4
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G30
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G33
Partial effect
1
0
−1
10
15
20
25
30
35
Temperature
Basis: TPRS

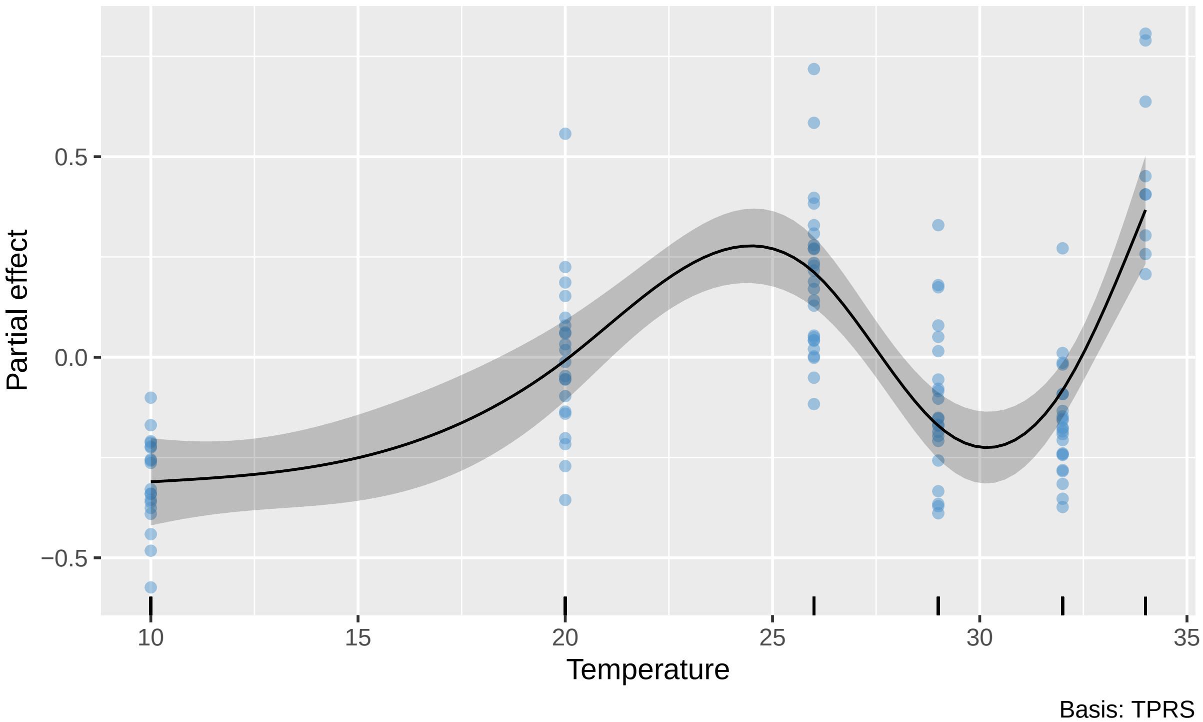

s(Temperature)
By: Line; G34
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

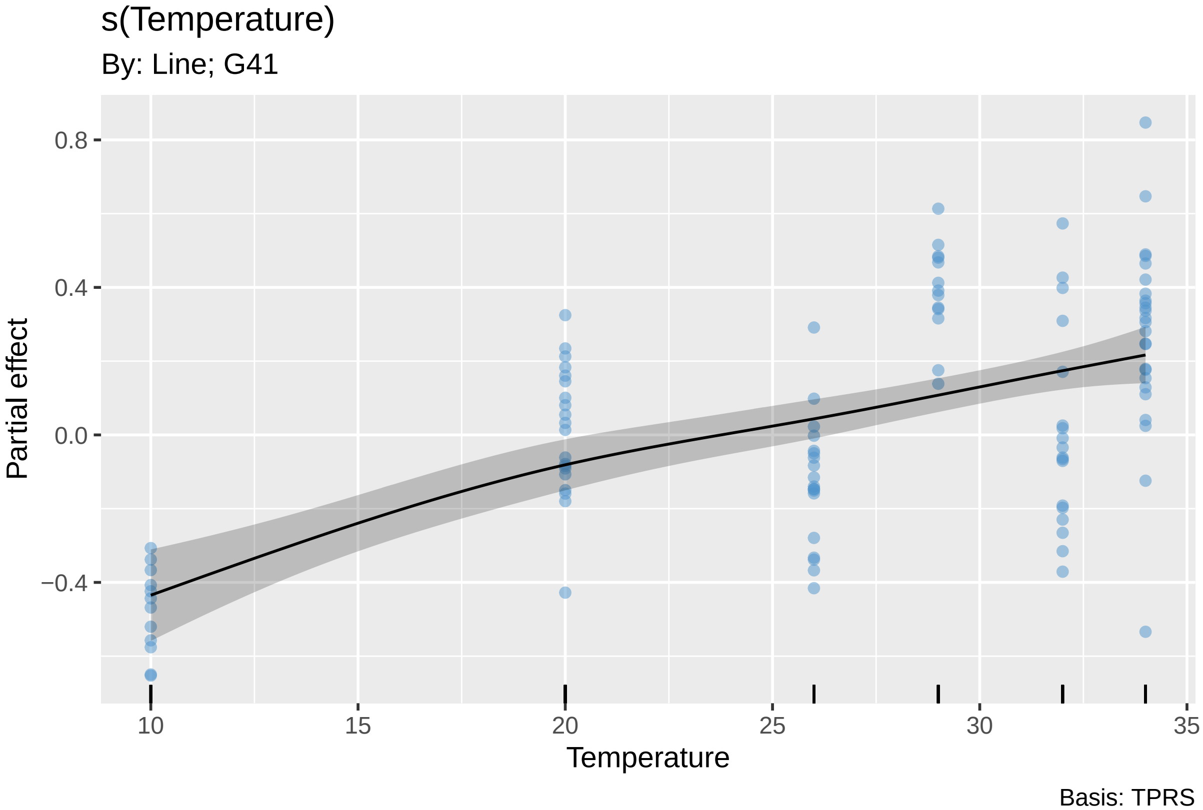

s(Temperature)
By: Line; G35
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G36
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

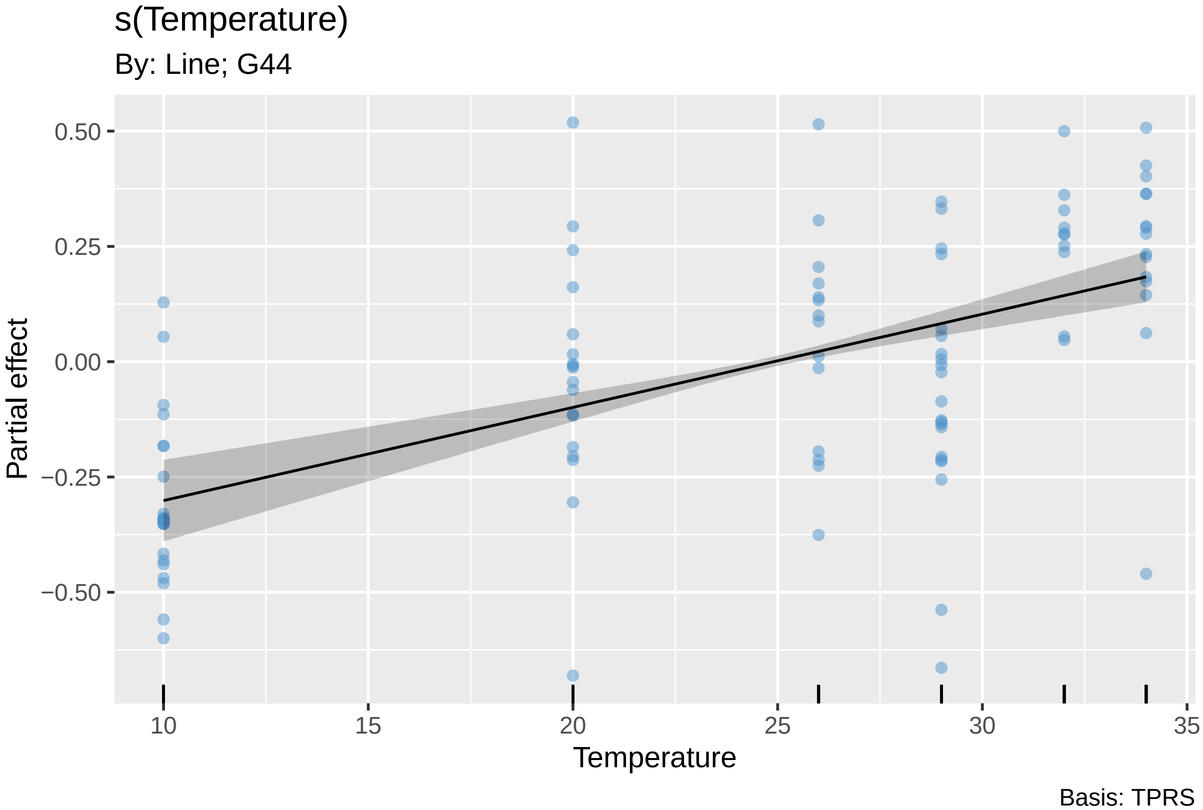
s(Temperature)

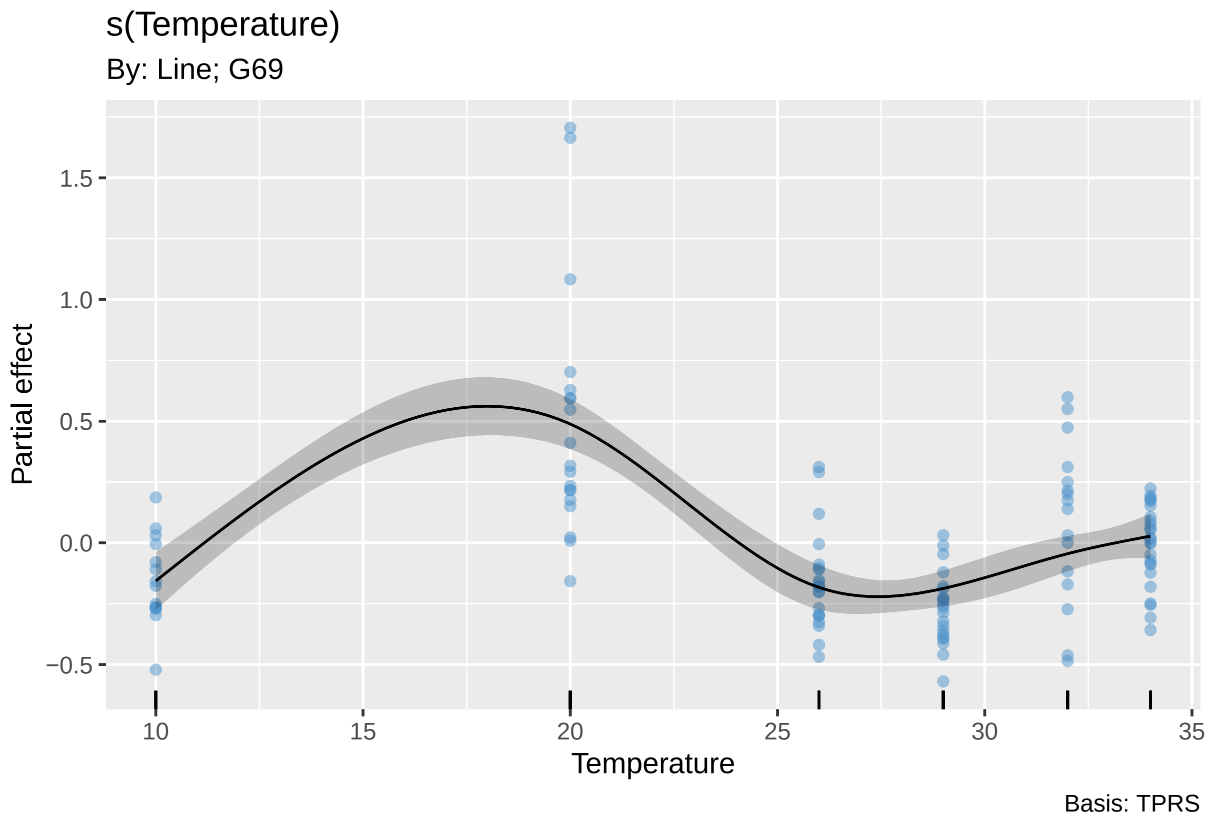
By: Line; G38

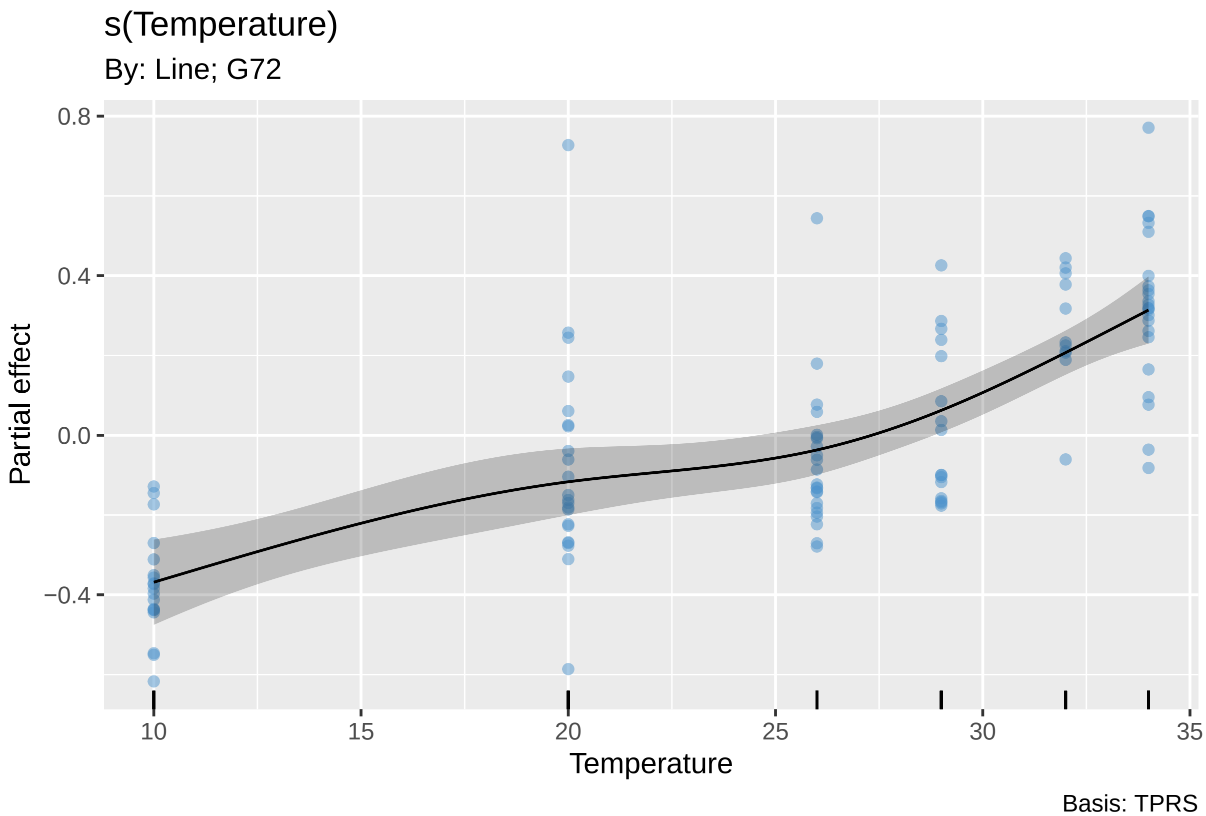




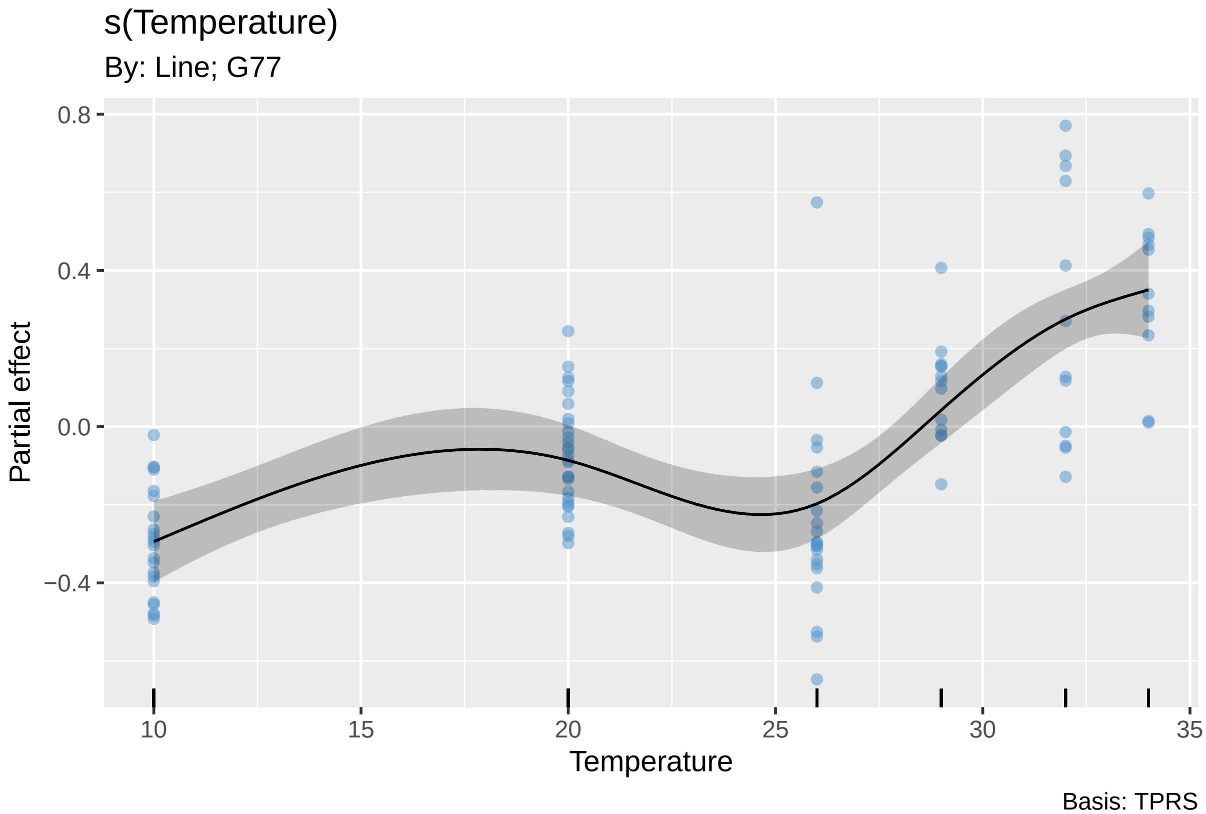
s(Temperature)
By: Line; G41
Partial effect
0.8
0.4
0.0
−0.4
10
15
20
25
30
35
Temperature
Basis: TPRS

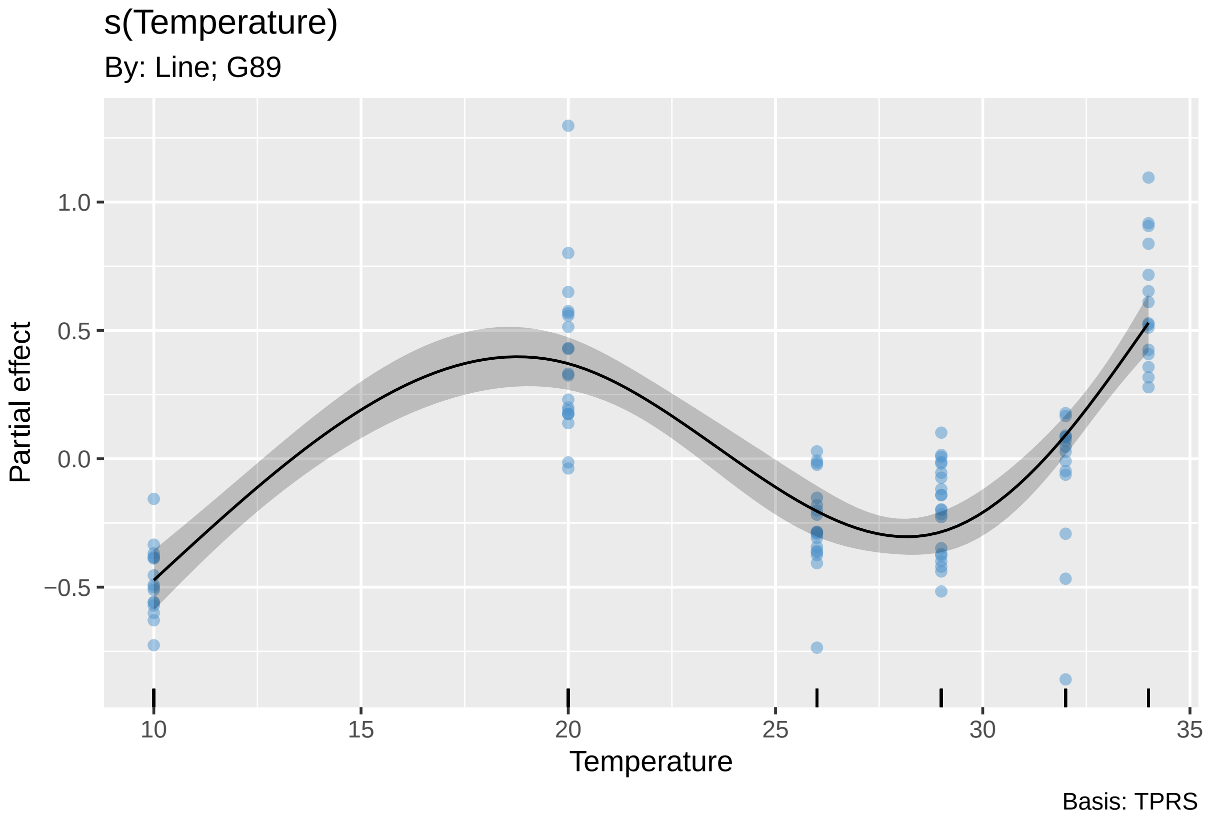

s(Temperature)
By: Line; G44
Partial effect
0.50
0.25
0.00
−0.25
−0.50
10
15
20
25
30
35
Temperature
Basis: TPRS

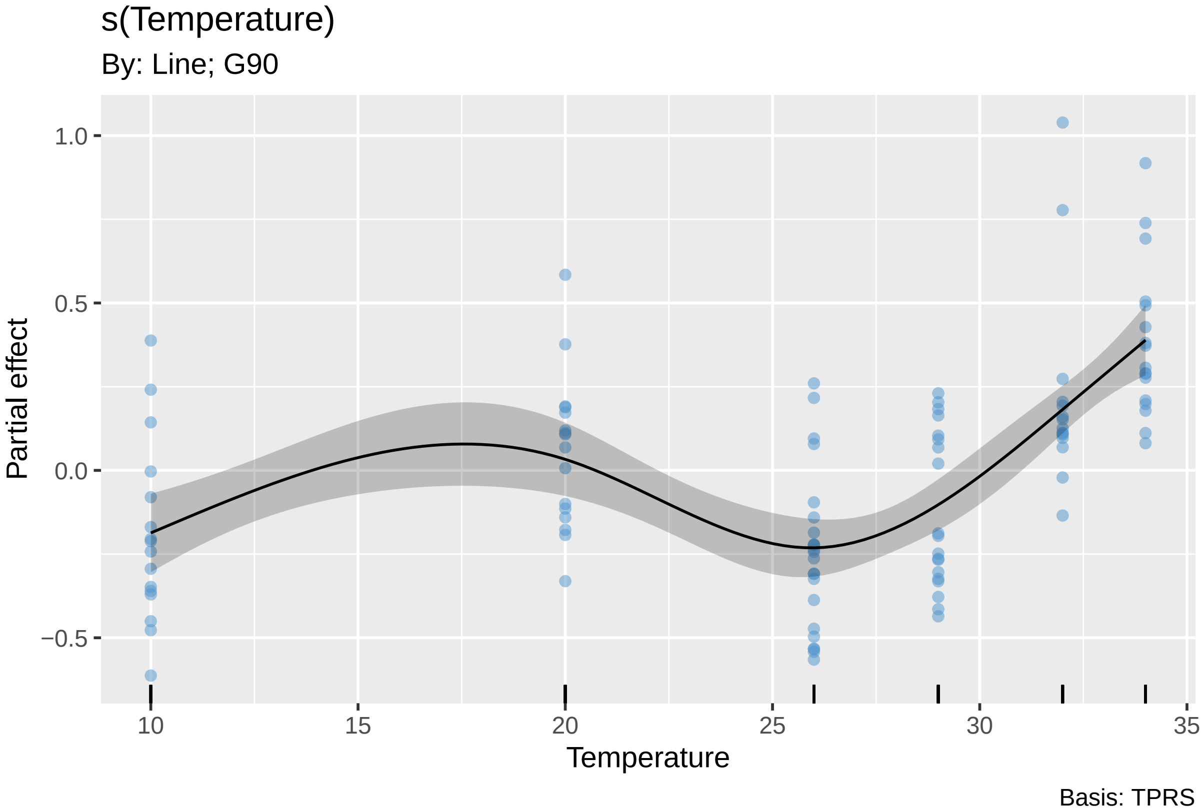

s(Temperature)
By: Line; G69
Partial effect
1.5
1.0
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS



s(Temperature)
By: Line; G72
Partial effect
0.8
0.4
0.0
−0.4
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G77
Partial effect
0.8
0.4
0.0
−0.4
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G89
Partial effect
1.0
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS



Again, we will assess whether the inclusion of the interaction terms improves the model's ability to predict the data.

```
gam_ar_noint <- gam(formula = mean_ar ~ Line + s(Temperature, k = 5, bs = 'tp'), data = morph_data, meth

AIC(gam_ar, gam_ar_noint)
```

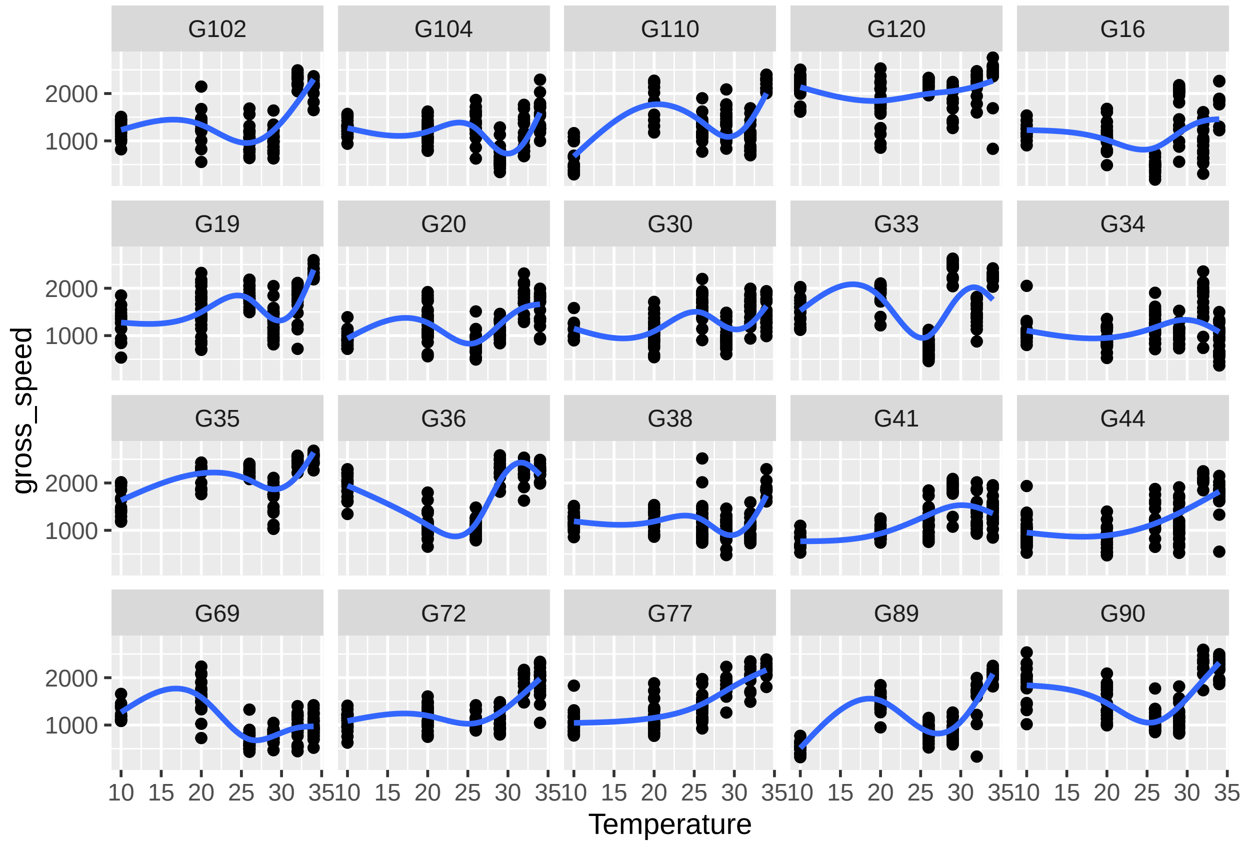
```
##                    df       AIC
## gam_ar       87.50686  409.7618
## gam_ar_noint 24.91808 1010.1726
```

And, again we see that AIC score is much lower for the model including the interaction (ΔAIC = 600.4108121).

Speed

For our final phenotype, we will look at speed.

```
ggplot(data = morph_data, aes(x = Temperature, y = gross_speed)) + geom_point() + geom_smooth(method =
  facet_wrap(facets = "Line", nrow = 4, ncol = 5)
```



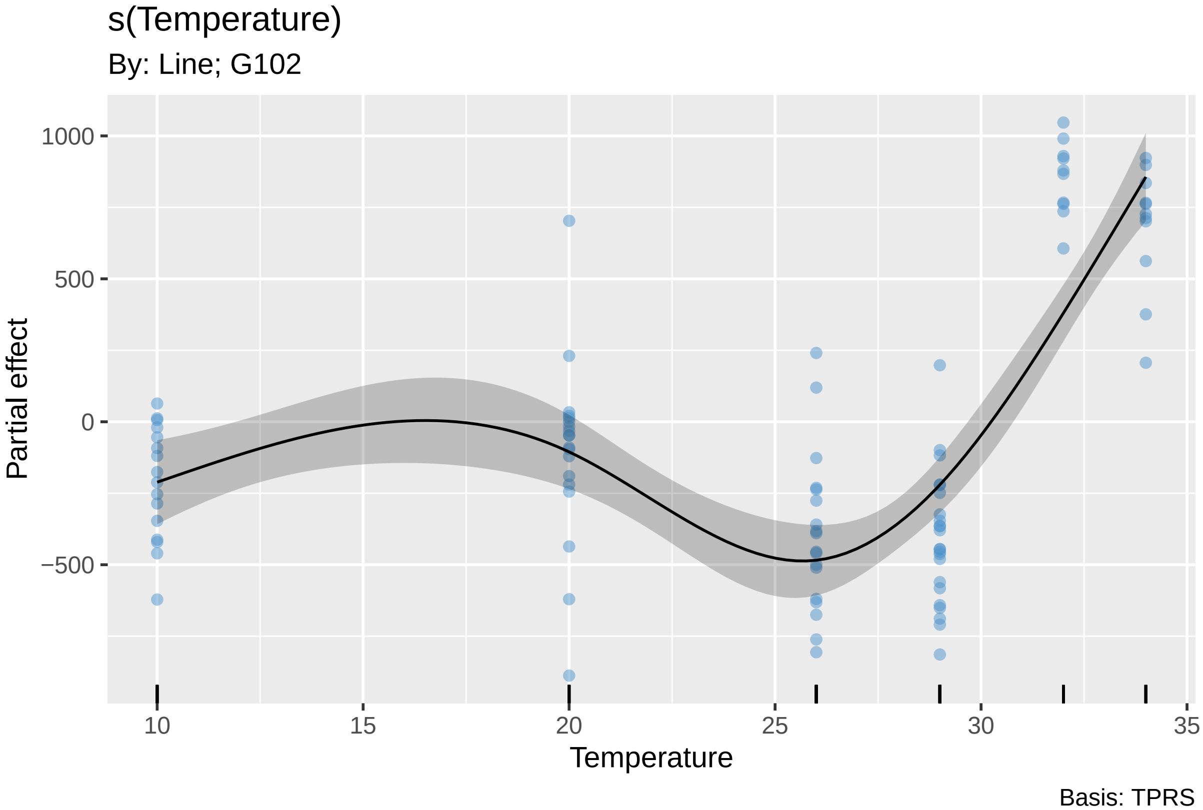
As for aspect ratio, we some general trends towards faster paramecium at higher temperatures with some exceptions. Again, we can fit a GAM with an interaction between temperature and line.

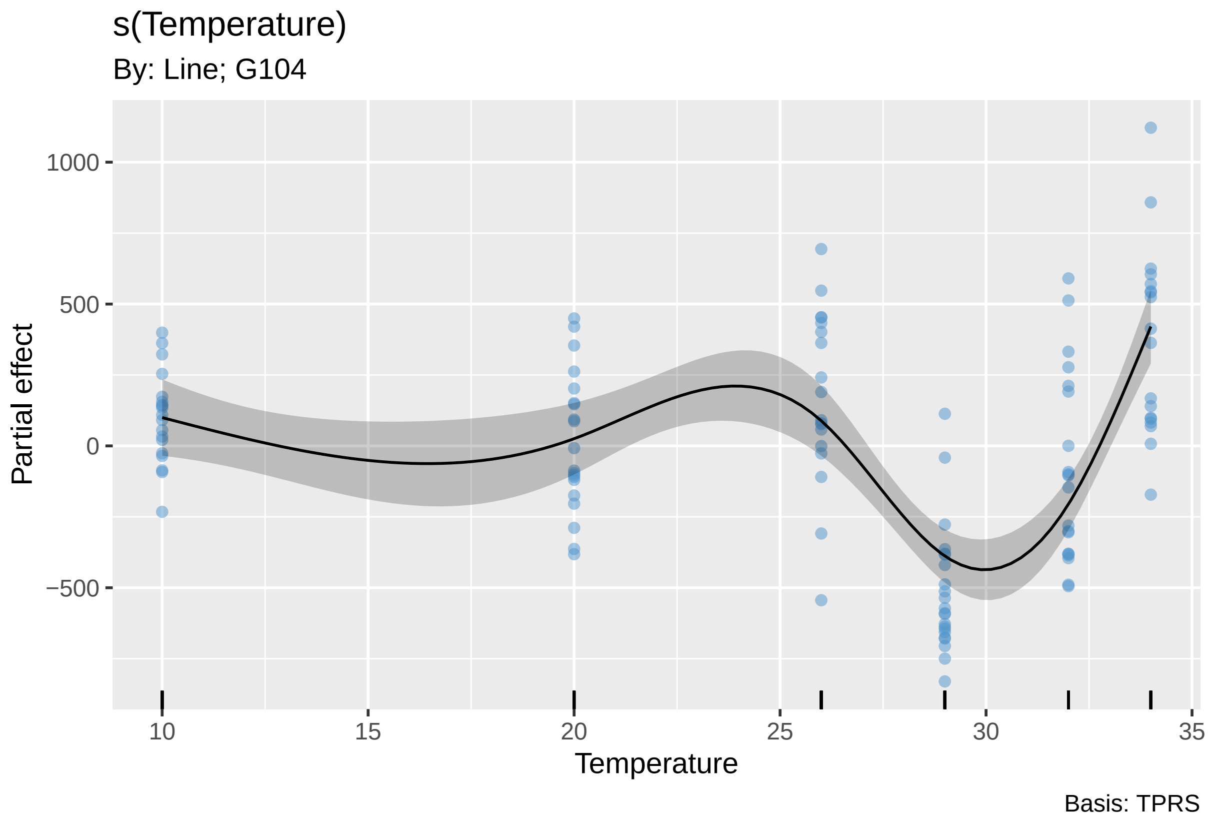
```
gam_speed <- gam(formula = gross_speed ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = mor

summary(gam_speed)
```

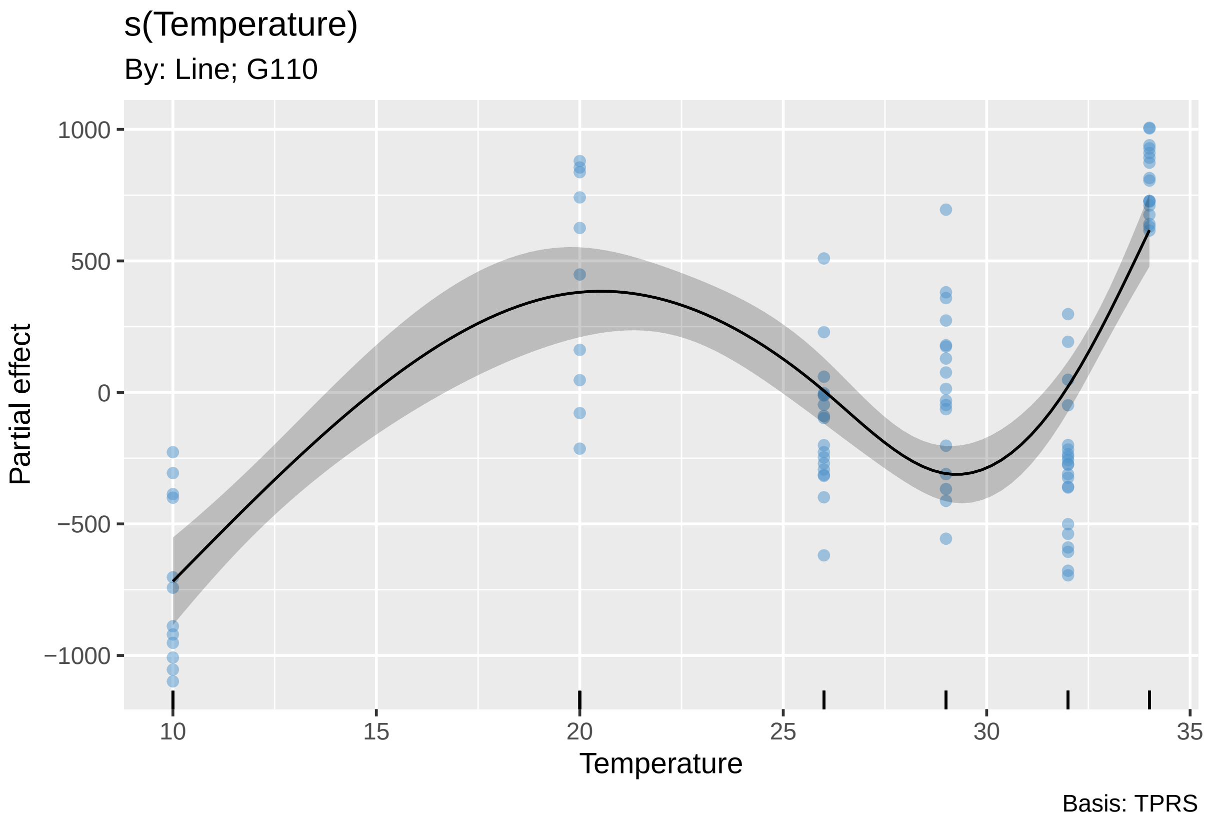
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## gross_speed ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1442.94      33.24  43.407  < 2e-16 ***
## LineG104      -272.12      44.83  -6.071 1.52e-09 ***
## LineG110       -51.81      48.12  -1.077 0.281794
## LineG120       621.35      47.36  13.119  < 2e-16 ***
## LineG16       -279.67      48.67  -5.746 1.05e-08 ***
## LineG19        194.27      45.62   4.258 2.16e-05 ***
## LineG20       -204.20      44.15  -4.625 3.99e-06 ***
## LineG30       -155.53      46.33  -3.357 0.000803 ***
## LineG33        170.81      46.84   3.647 0.000273 ***
## LineG34       -297.59      45.51  -6.538 7.86e-11 ***
## LineG35        642.37      49.86  12.884  < 2e-16 ***
```

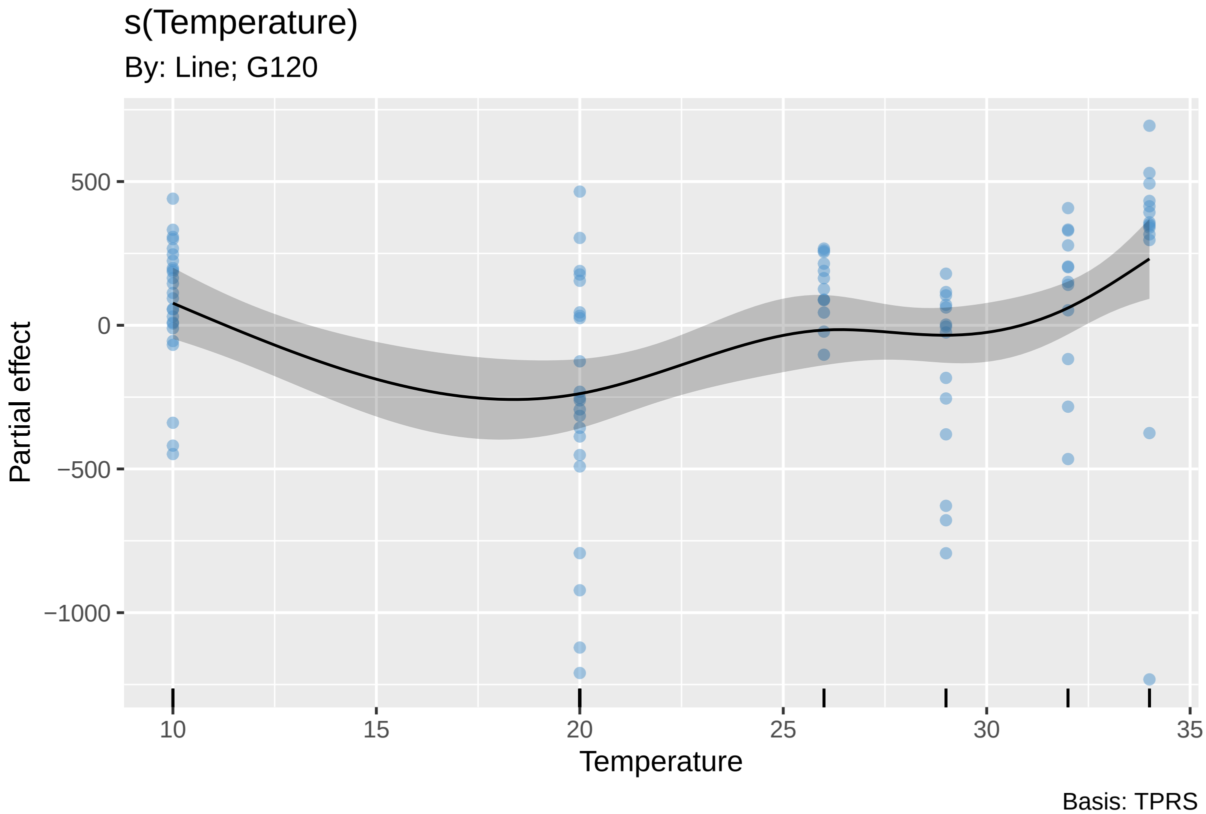
```
## LineG36       314.54      48.12   6.536 7.98e-11 ***
## LineG38      -221.66      44.98  -4.928 9.00e-07 ***
## LineG41      -220.13      46.29  -4.756 2.12e-06 ***
## LineG44      -169.46      46.57  -3.639 0.000281 ***
## LineG69      -397.45      44.83  -8.867  < 2e-16 ***
## LineG72       -96.56      45.11  -2.141 0.032427 *
## LineG77       119.42      46.40   2.573 0.010142 *
## LineG89      -227.60      45.84  -4.964 7.47e-07 ***
## LineG90       188.88      45.90   4.115 4.03e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                             edf Ref.df      F  p-value
## s(Temperature):LineG102 3.705  3.935 37.815  < 2e-16 ***
## s(Temperature):LineG104 3.929  3.996 18.331  < 2e-16 ***
## s(Temperature):LineG110 3.877  3.989 37.044  < 2e-16 ***
## s(Temperature):LineG120 3.327  3.687  5.748 0.000198 ***
## s(Temperature):LineG16  3.821  3.976 13.094  < 2e-16 ***
## s(Temperature):LineG19  3.935  3.997 26.189  < 2e-16 ***
## s(Temperature):LineG20  3.825  3.976 27.344  < 2e-16 ***
## s(Temperature):LineG30  3.880  3.989 10.013  < 2e-16 ***
## s(Temperature):LineG33  3.949  3.998 34.318  < 2e-16 ***
## s(Temperature):LineG34  3.531  3.845  6.037 0.001777 **
## s(Temperature):LineG35  3.725  3.943 18.750  < 2e-16 ***
## s(Temperature):LineG36  3.918  3.995 56.109  < 2e-16 ***
## s(Temperature):LineG38  3.883  3.990 10.760  < 2e-16 ***
## s(Temperature):LineG41  3.336  3.713 19.569  < 2e-16 ***
## s(Temperature):LineG44  2.755  3.176 34.053  < 2e-16 ***
## s(Temperature):LineG69  3.813  3.974 24.928  < 2e-16 ***
## s(Temperature):LineG72  3.468  3.800 36.726  < 2e-16 ***
## s(Temperature):LineG77  2.744  3.160 49.696  < 2e-16 ***
## s(Temperature):LineG89  3.848  3.982 63.351  < 2e-16 ***
## s(Temperature):LineG90  3.714  3.941 46.158  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.634   Deviance explained =   65%
## -REML =  15026  Scale est. = 1.0449e+05  n = 2100
```

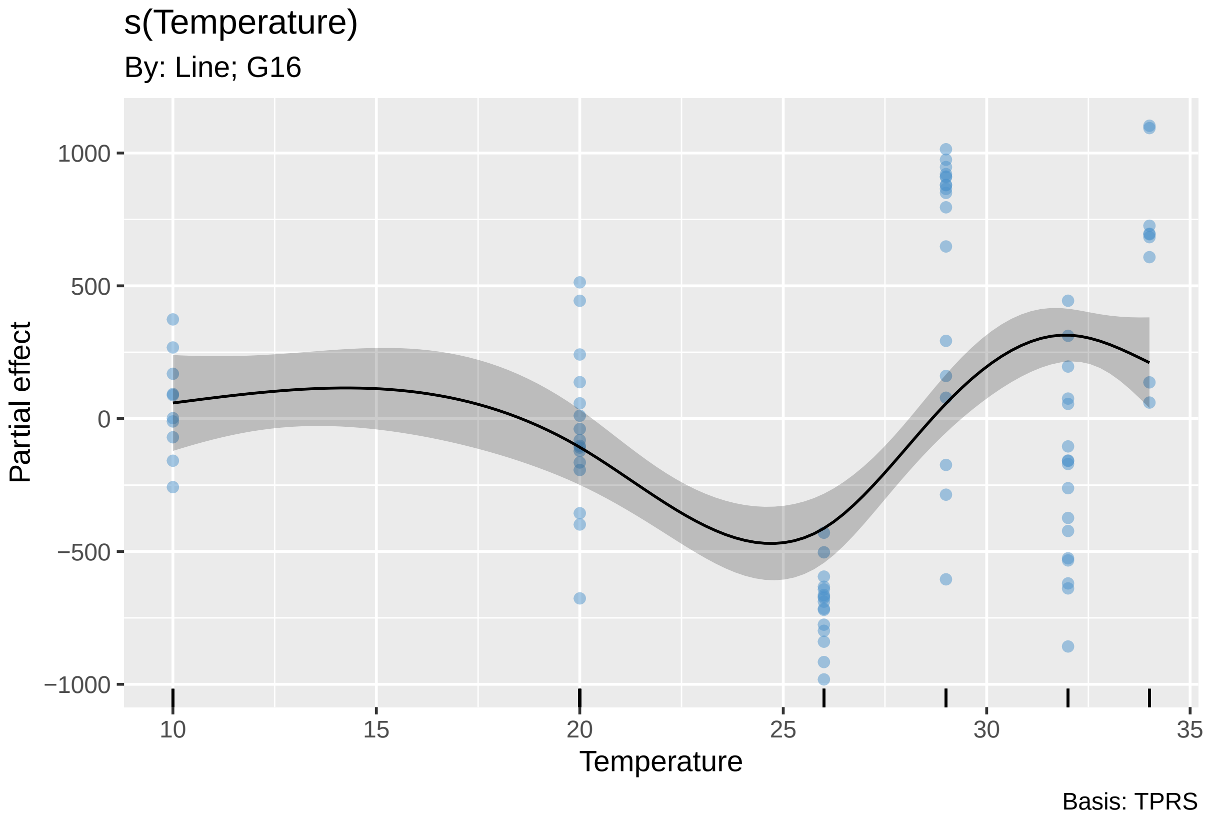
```r
for (i in 1:length(smooths(gam_speed))) {
  out <- draw(gam_speed, select = i, residuals = TRUE)
  print(out) }
```

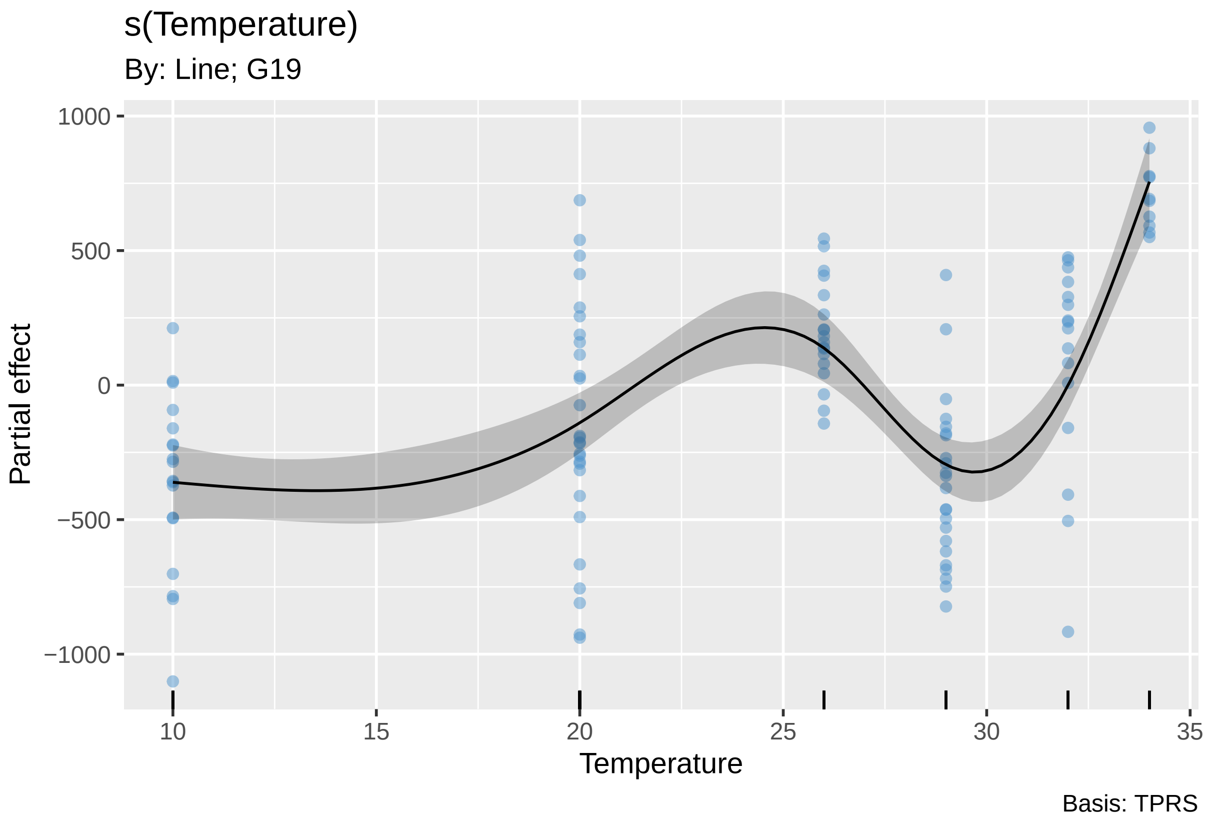

s(Temperature)
By: Line; G102
1000
500
0
−500
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

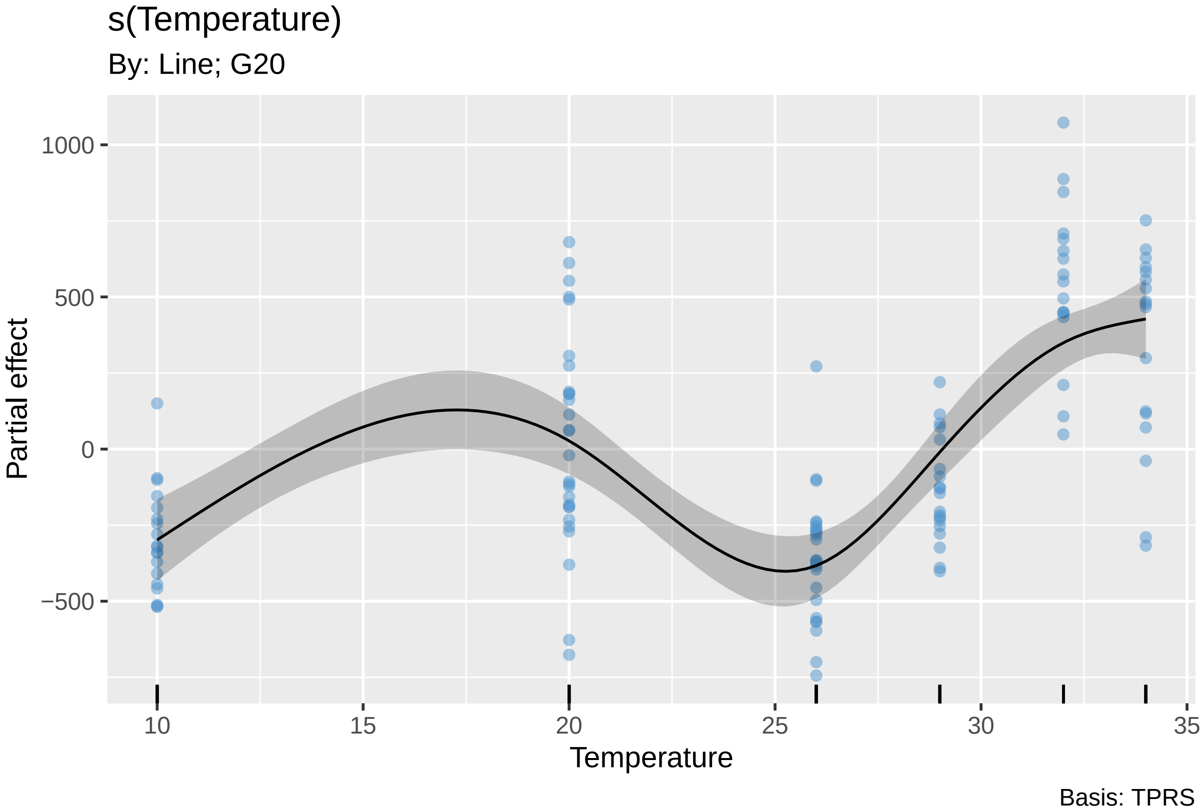

s(Temperature)
By: Line; G104
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

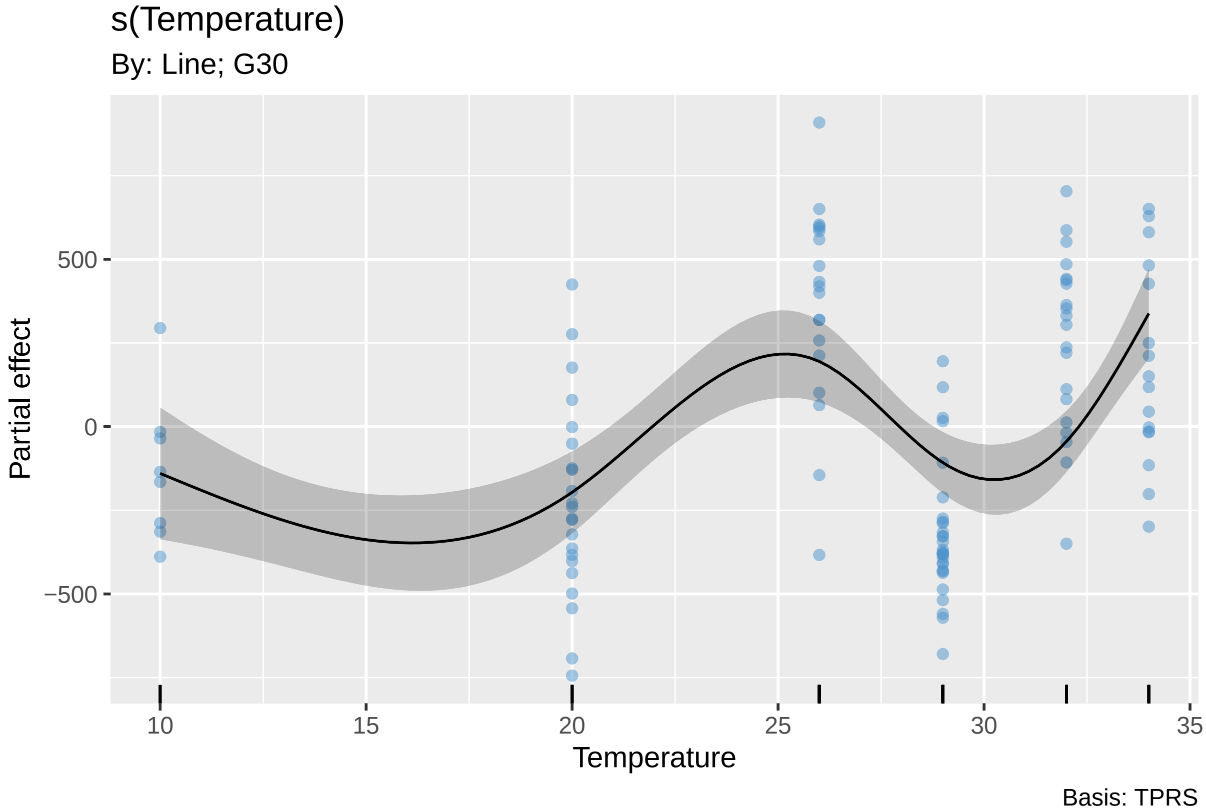

s(Temperature)
By: Line; G110
Partial effect
1000
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

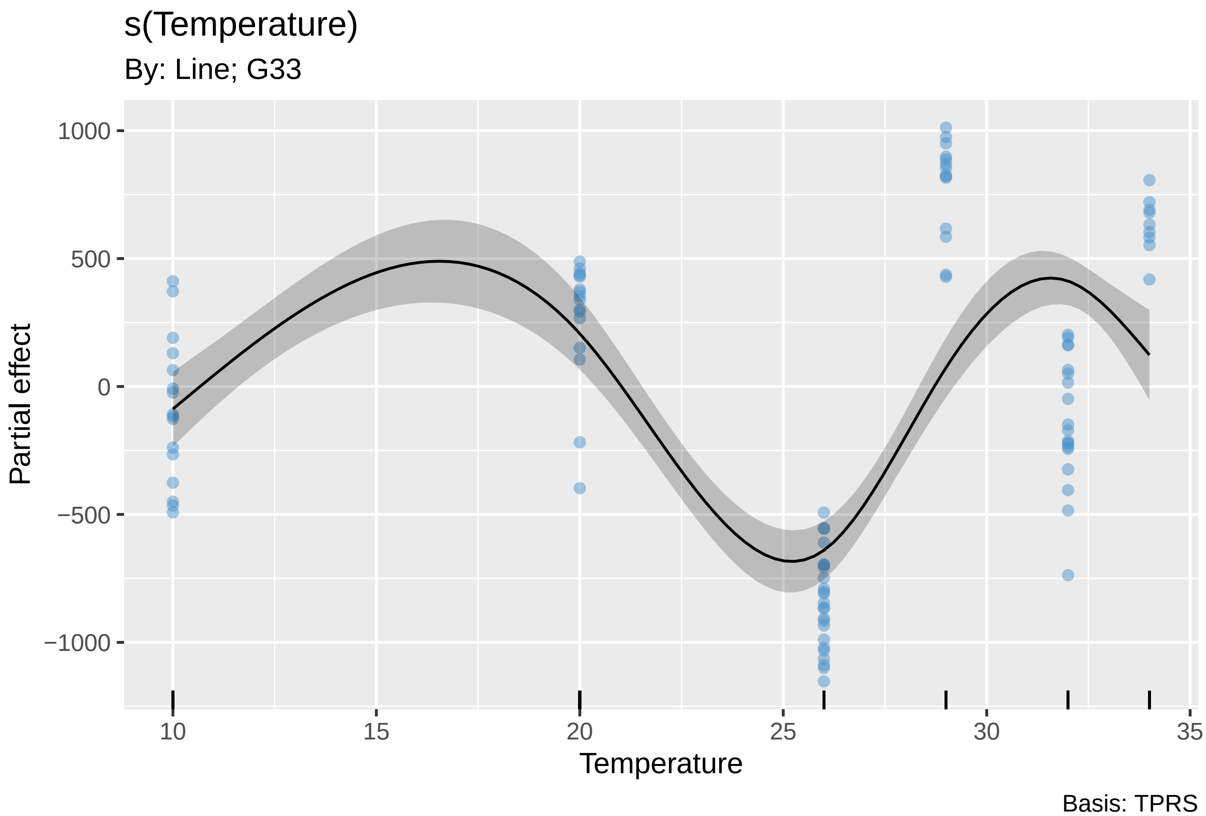

s(Temperature)
By: Line; G120
Partial effect
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

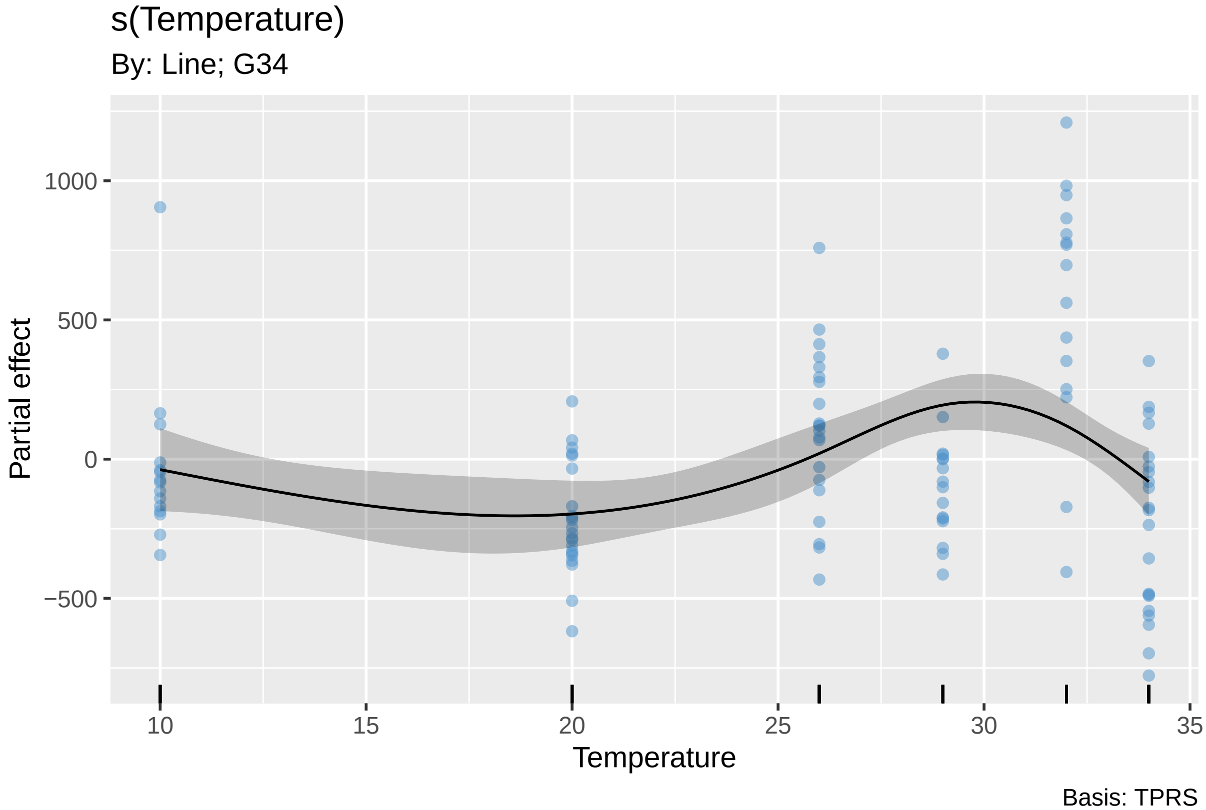

s(Temperature)
By: Line; G16
Partial effect
1000
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

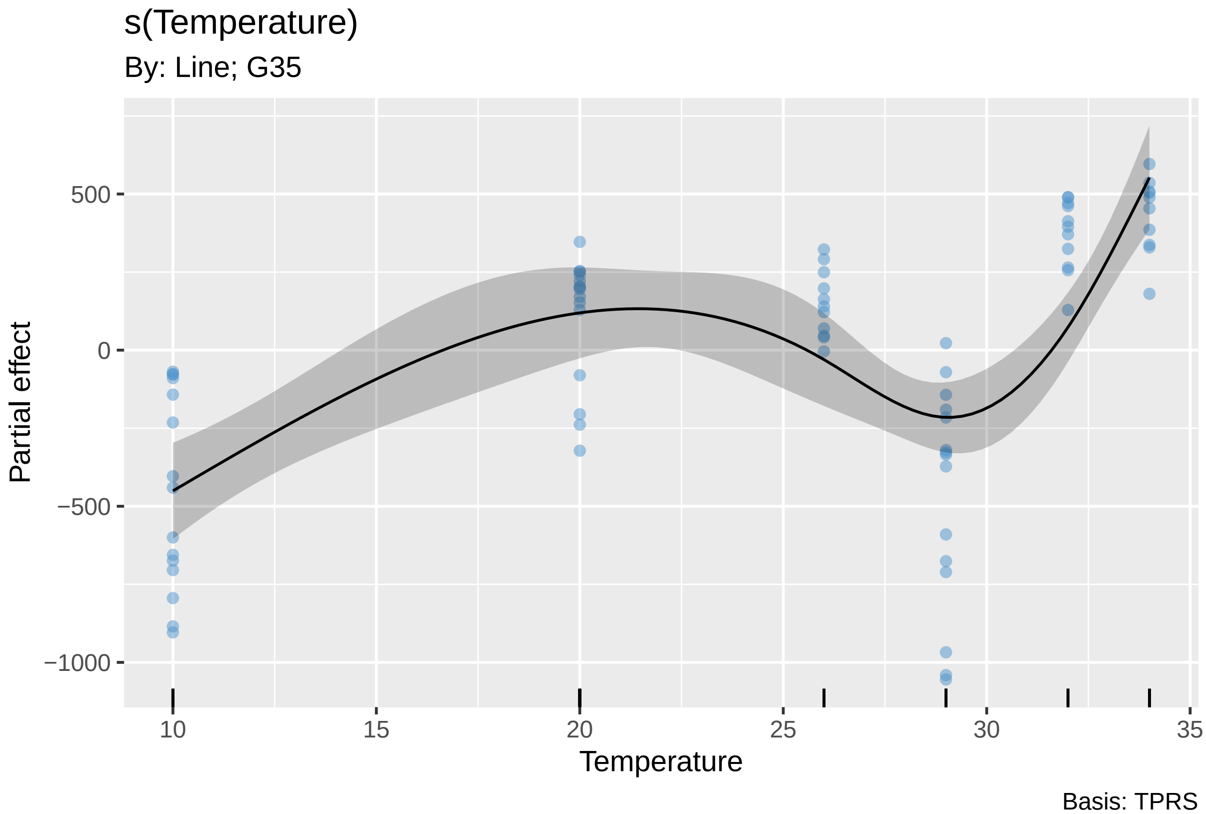

s(Temperature)
By: Line; G19
Partial effect
1000
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

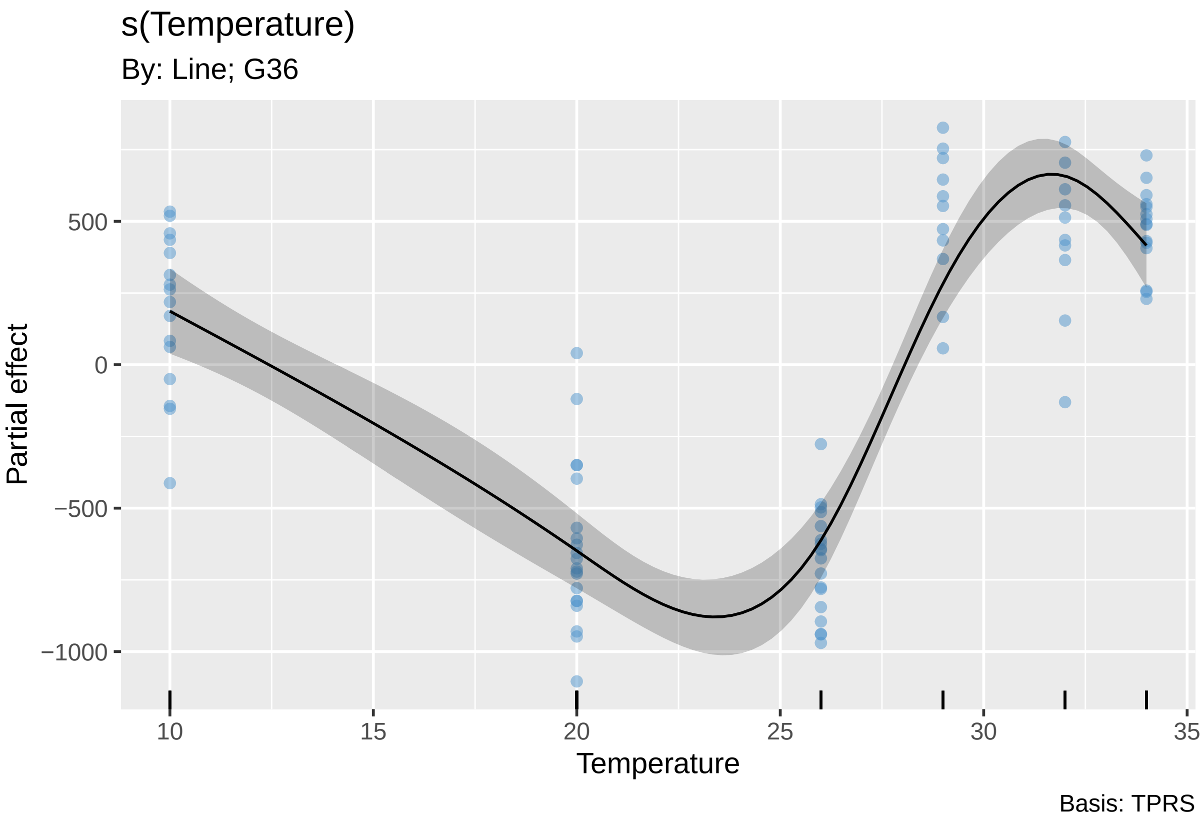

s(Temperature)
By: Line; G20
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

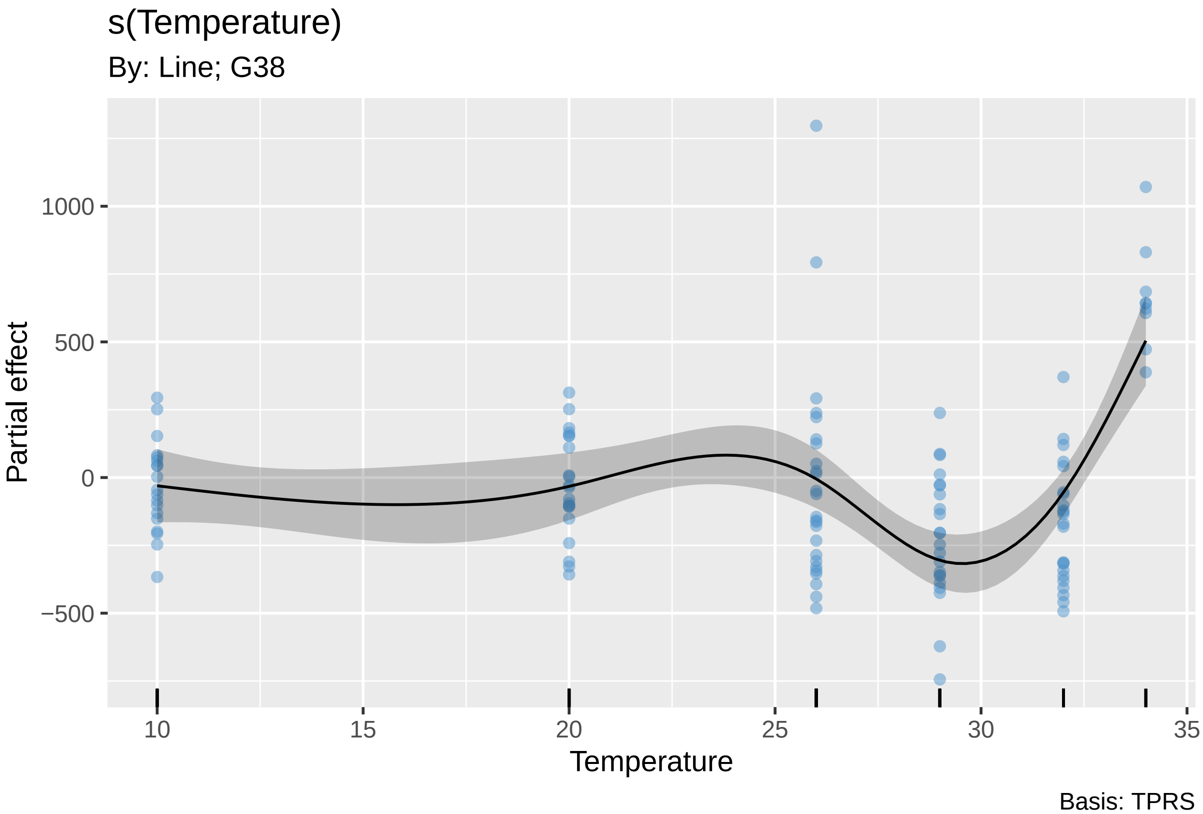

s(Temperature)
By: Line; G30
Partial effect
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

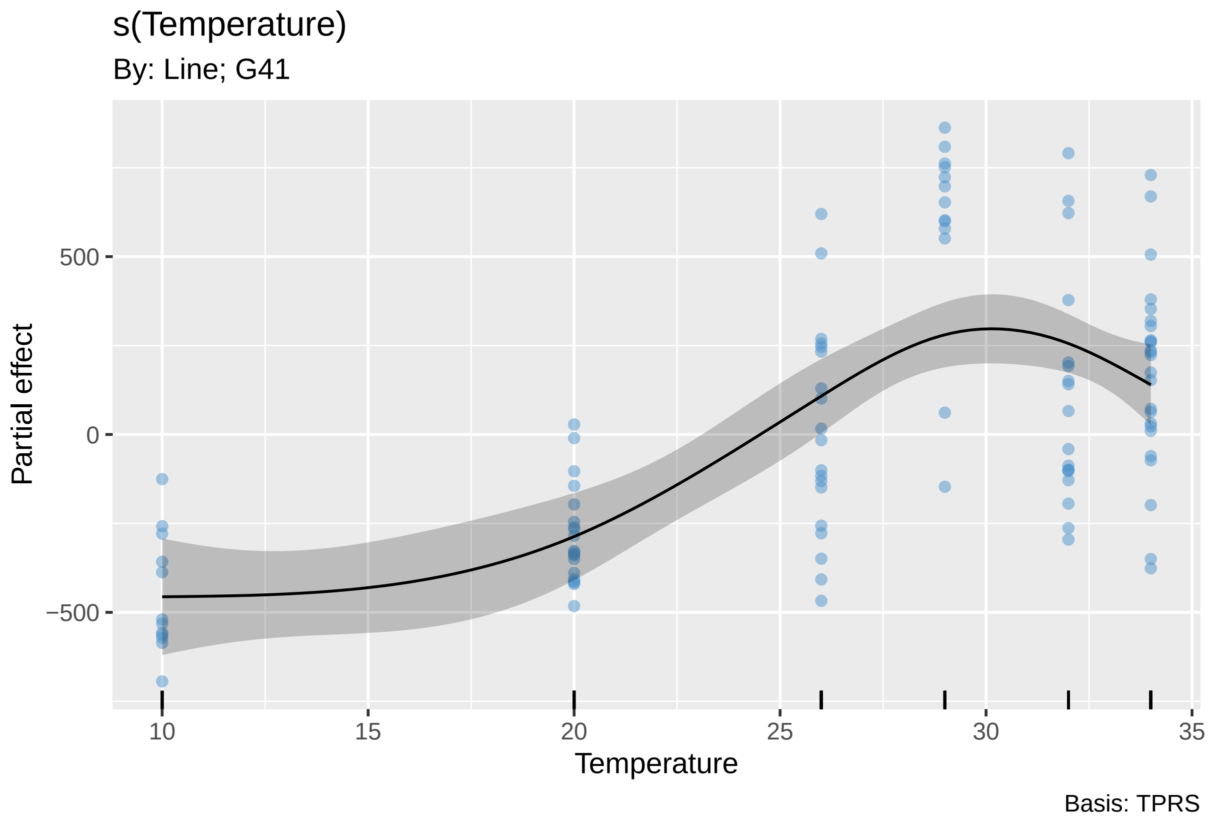

s(Temperature)
By: Line; G33
1000
500
0
−500
−1000
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

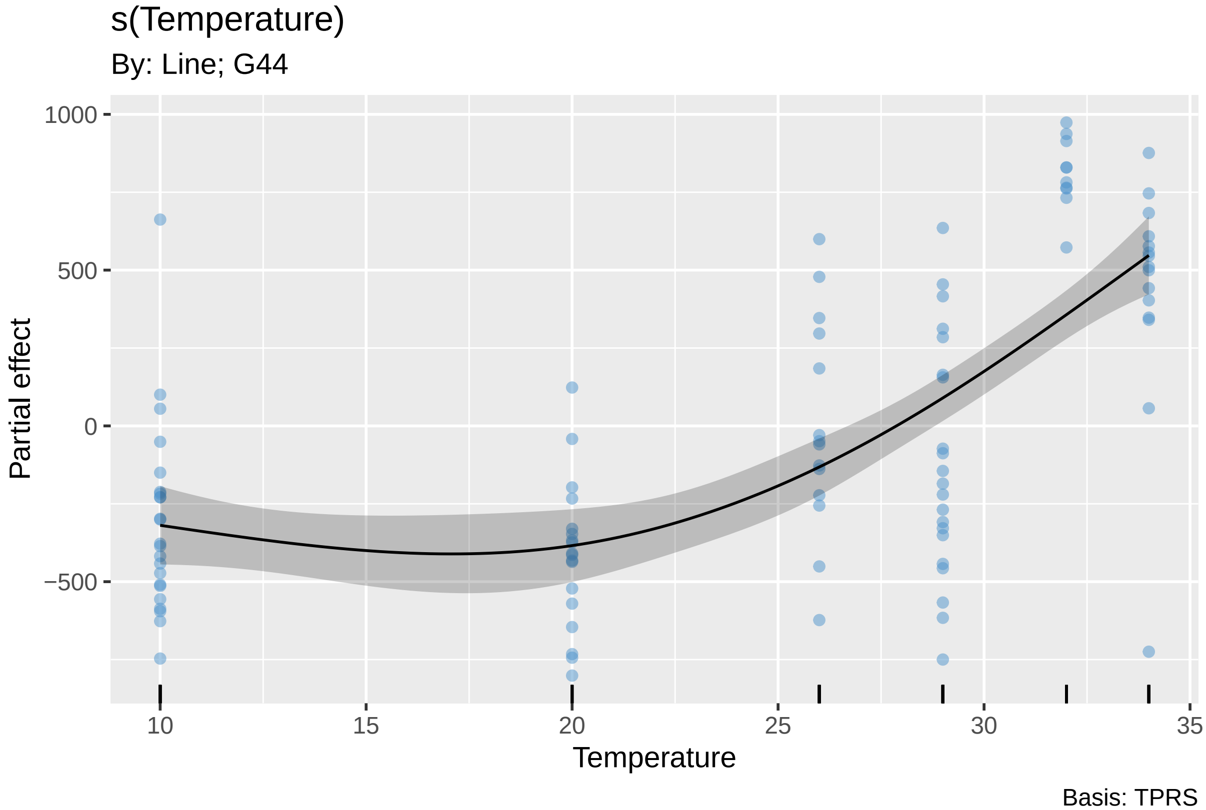

s(Temperature)
By: Line; G34
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

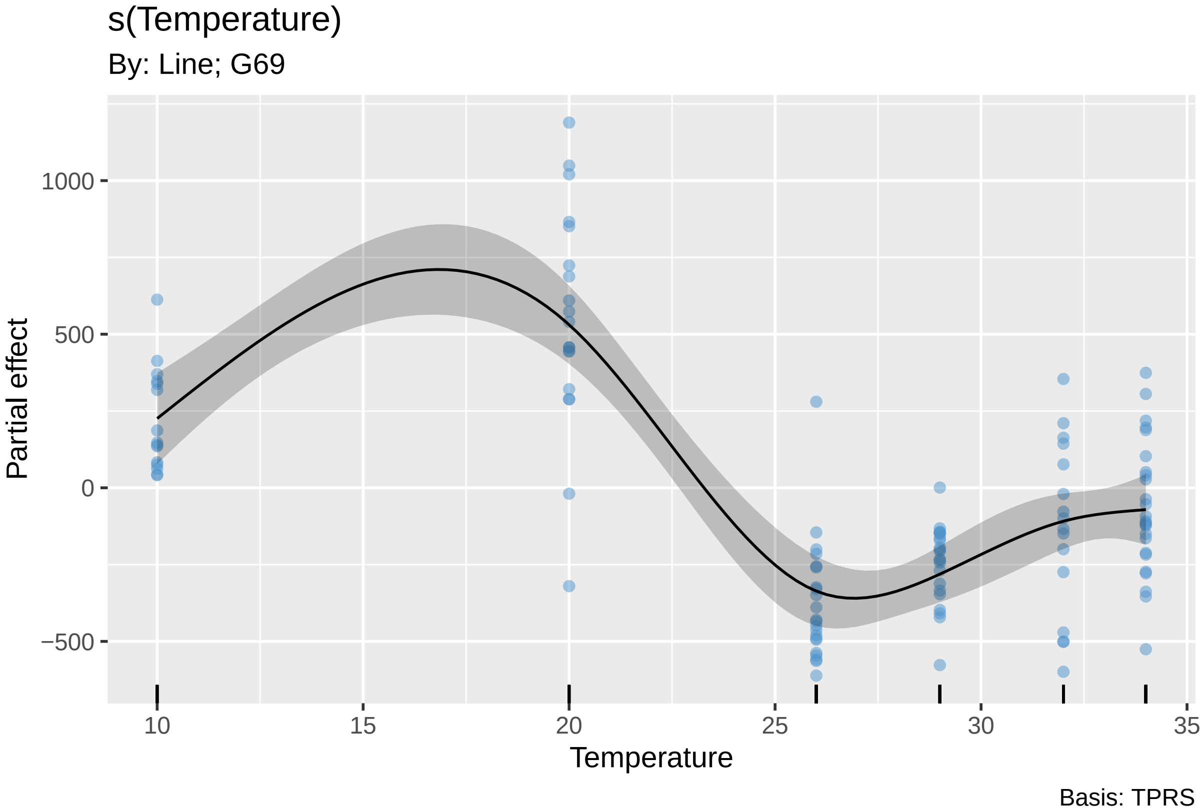

s(Temperature)
By: Line; G35
Partial effect
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

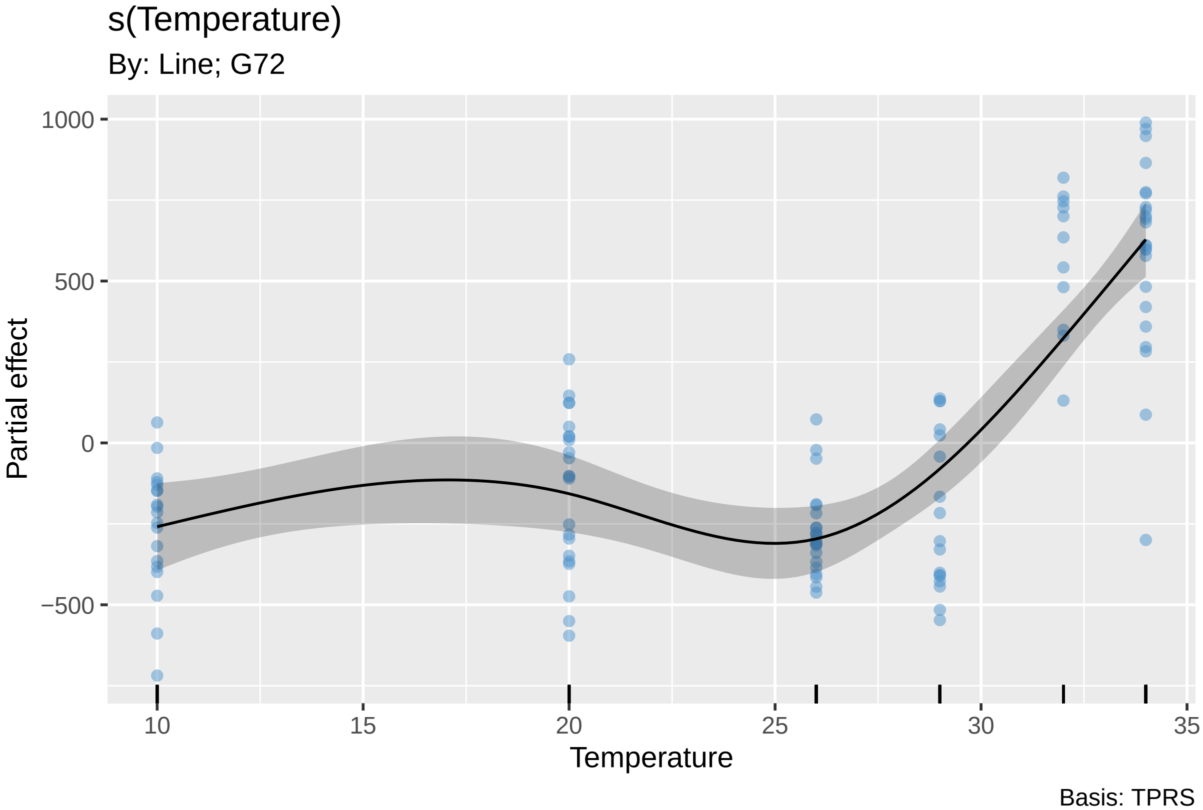

s(Temperature)
By: Line; G36
Partial effect
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

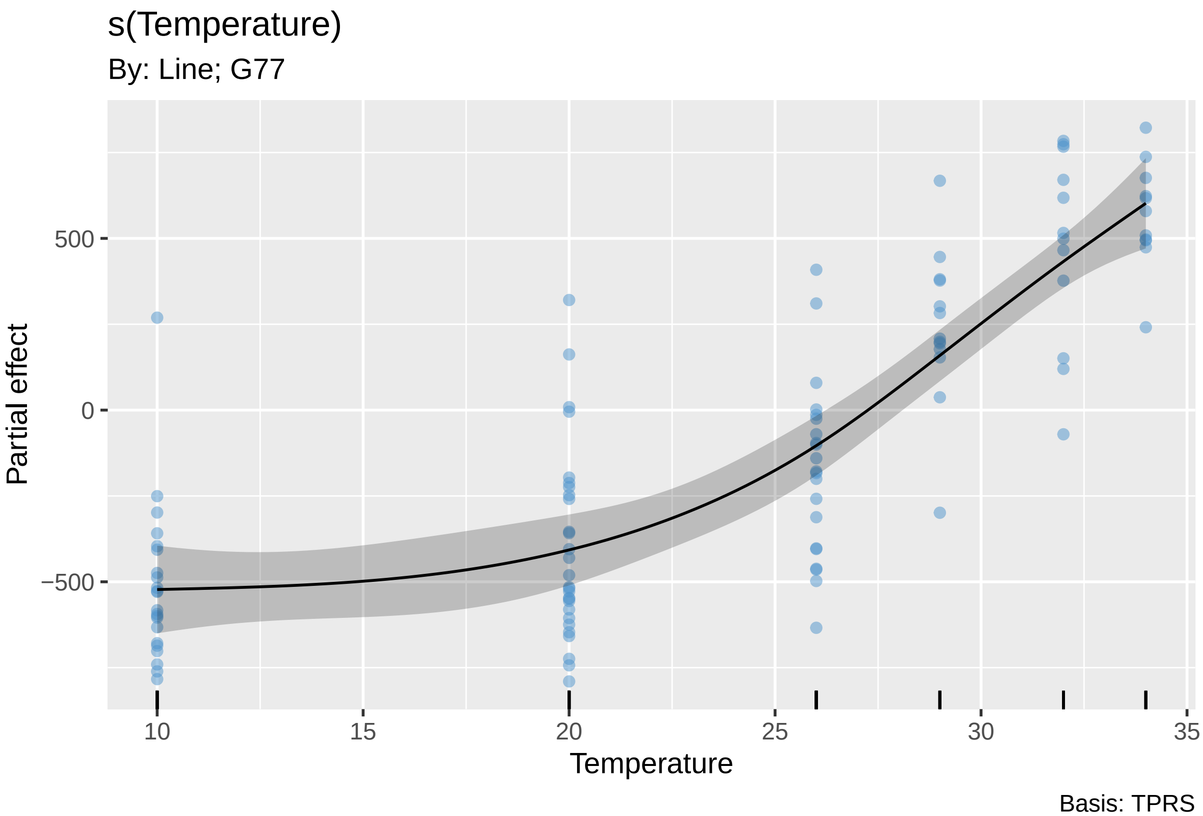

s(Temperature)
By: Line; G38
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G41
Partial effect
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

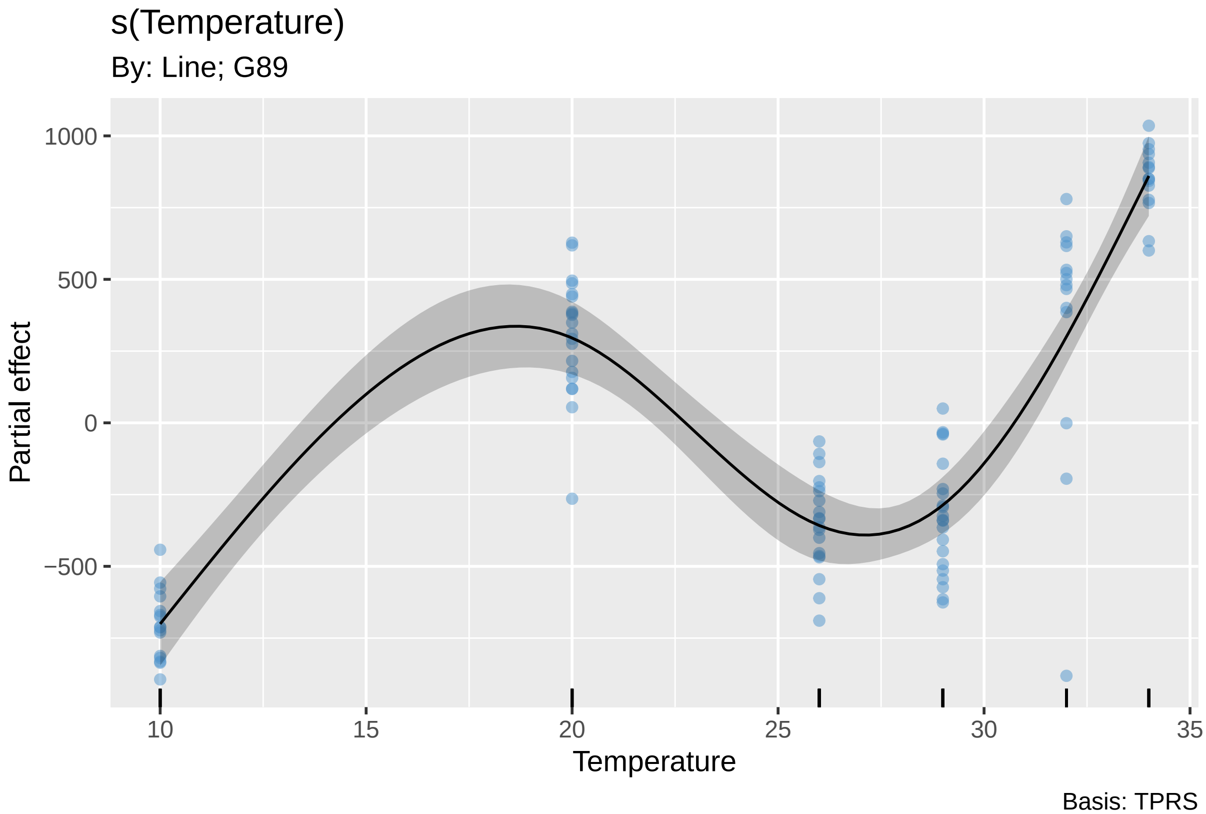

s(Temperature)
By: Line; G44
1000
500
0
−500
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

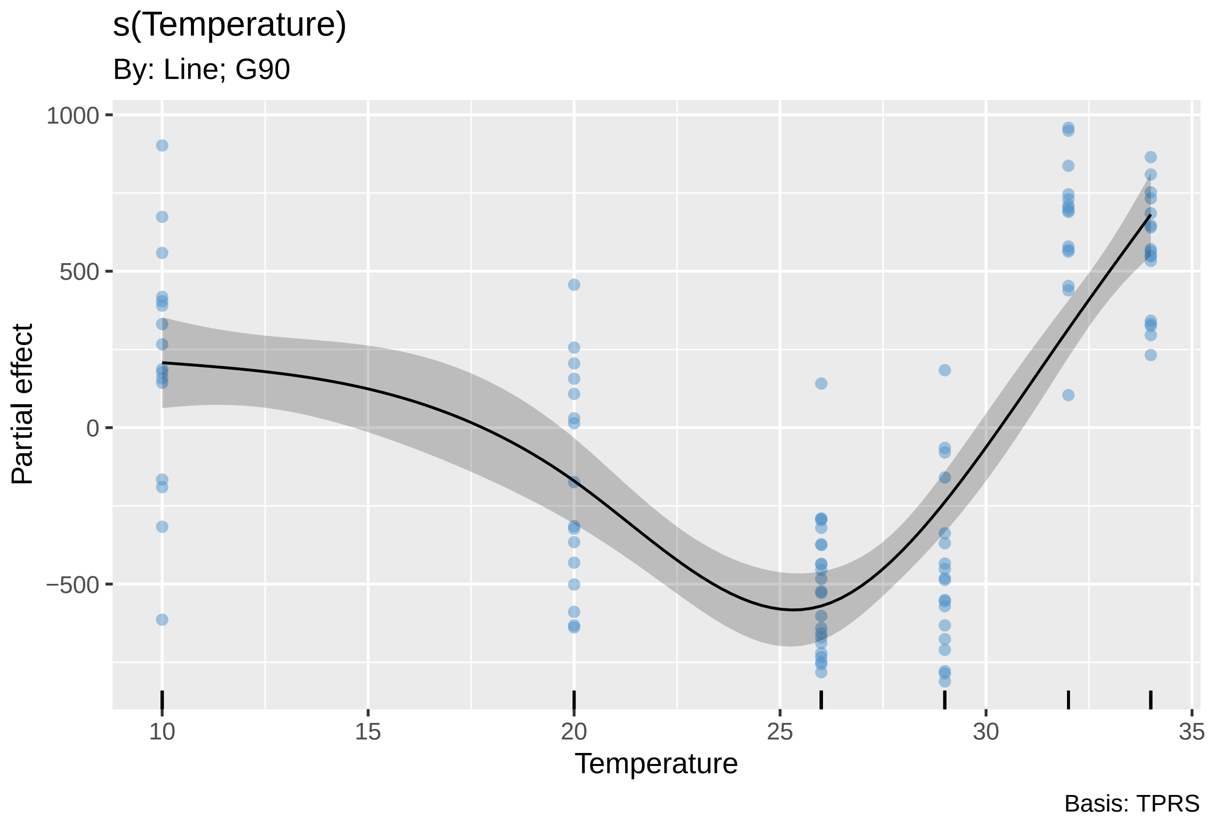

s(Temperature)
By: Line; G69
1000
500
0
−500
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G72
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS



s(Temperature)
By: Line; G77
Partial effect
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G89
1000
500
0
−500
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS



Again, we will assess whether the inclusion of the interaction terms improves the model's ability to predict the data.

```
gam_speed_noint <- gam(formula = gross_speed ~ Line + s(Temperature, k = 5, bs = 'tp'), data = morph_da

AIC(gam_speed, gam_speed_noint)
```

```
##                        df      AIC
## gam_speed       98.06386 30330.04
## gam_speed_noint 24.89081 31271.69
```

And, again we see that AIC score is much lower for the model including the interaction (ΔAIC = 941.6469014).

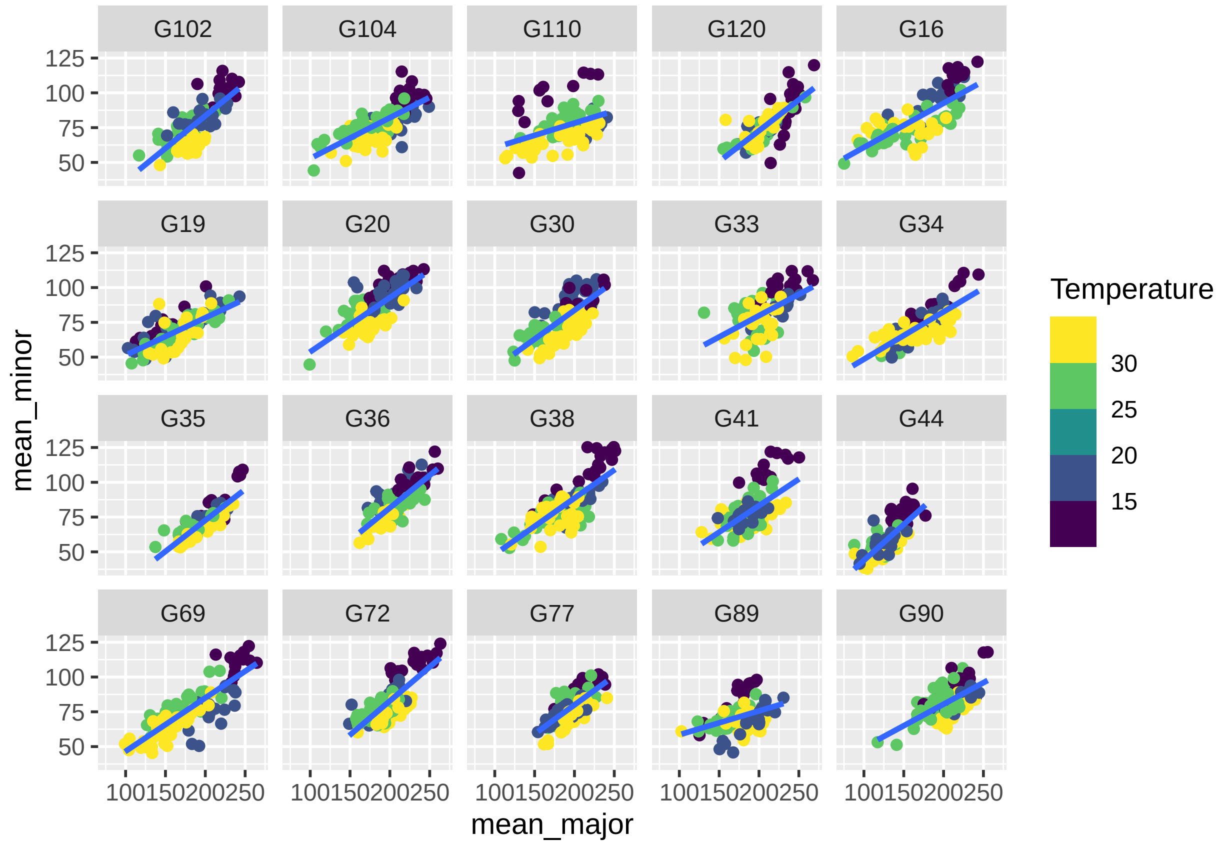
Examining the structure of morphological plasticity in Paramecium

From our previous results, we know that genetic variation in the paramecia is present for size, aspect ratio, and speed. We also know that variation in length and width are correlated and aspect ratio and speed, but these are not correlated with one another. One question we might be interested in is whether plasticity follows a similar pattern? To examine this, we can look at some plots for each outcrossed line that assess the relationship within each line between the morphological and movement variables and temperature.

First, we can look at length and width.

```
ggplot(data = morph_data, aes(x = mean_major, y = mean_minor)) + geom_point(aes(color = Temperature)) +
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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

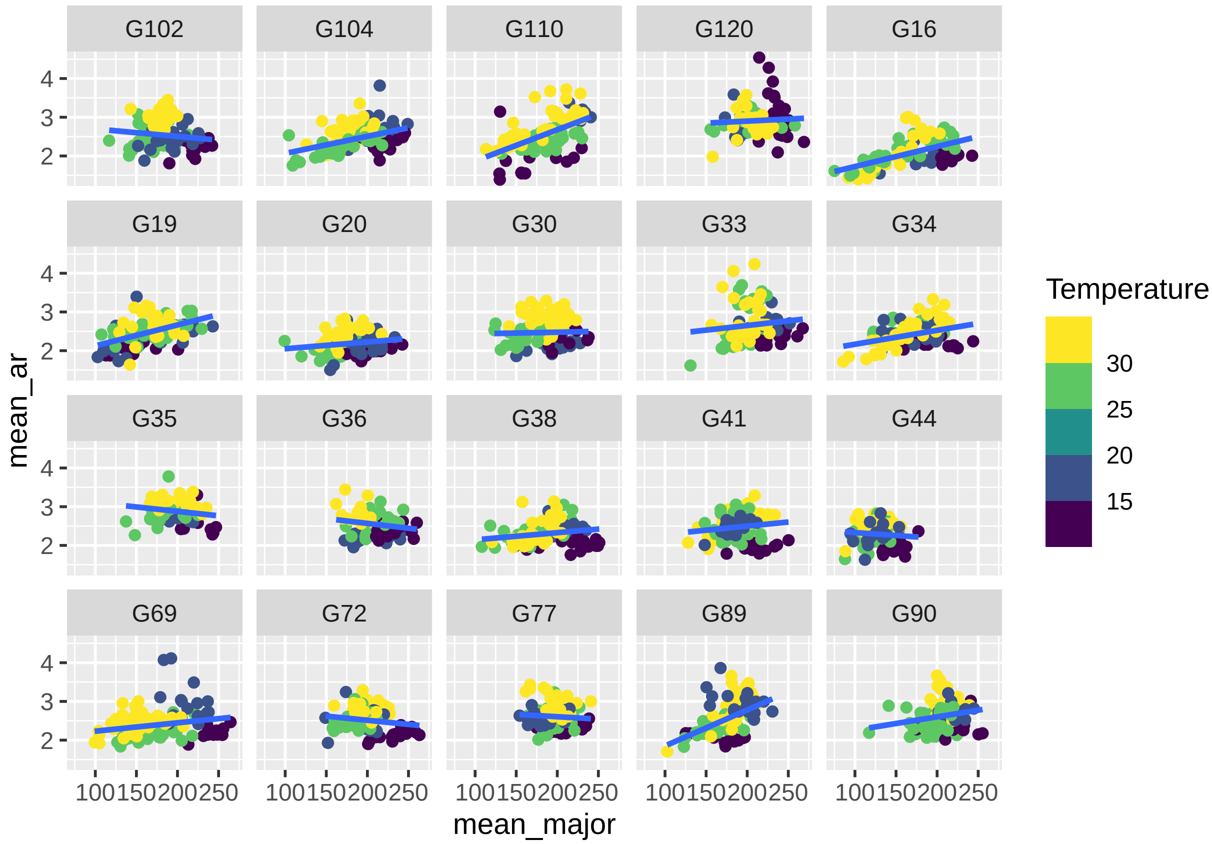


Across lines, it does appear to be the case that shorter cells are also less wide. Furthermore, looking at the relationships with temperature, many of the lines tend to have the longest and widest cells at low temperatures and the shortest, thinnest cells at high temperatures (e.g. G69). There also is some evidence here that the degree of plasticity differs across the outcrossed lines. For example, lines such as G69 and G16 span a large range of the x-axis across temperatures, whereas, lines such as G77 and G44 occupy a shorter range along the x-axis.

Now let's look at length and its relationship with aspect ratio.

```
ggplot(data = morph_data, aes(x = mean_major, y = mean_ar)) + geom_point(aes(color = Temperature)) + ge
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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

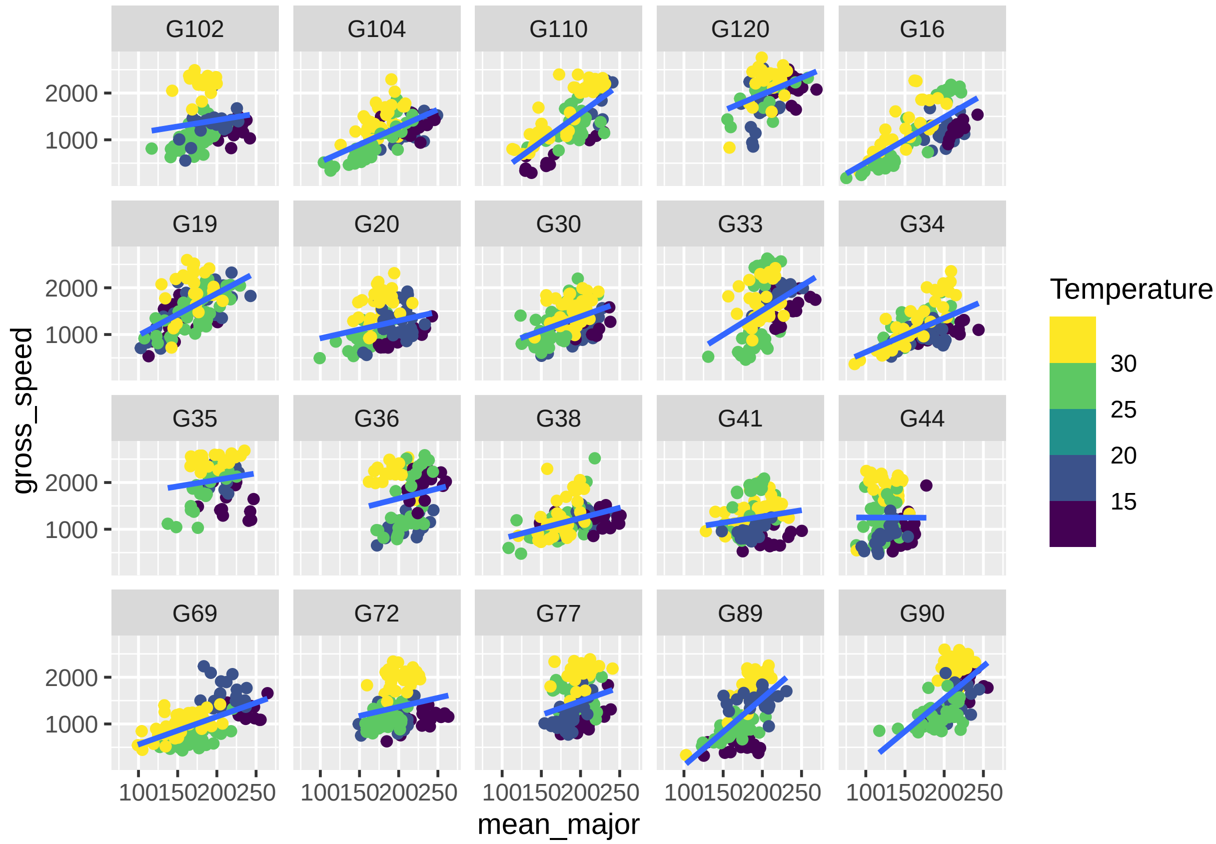


Here, we see generally shallower slopes in the relationship between length and aspect ratio.

Let's also look at length and speed.

```
ggplot(data = morph_data, aes(x = mean_major, y = gross_speed)) + geom_point(aes(color = Temperature)) +
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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

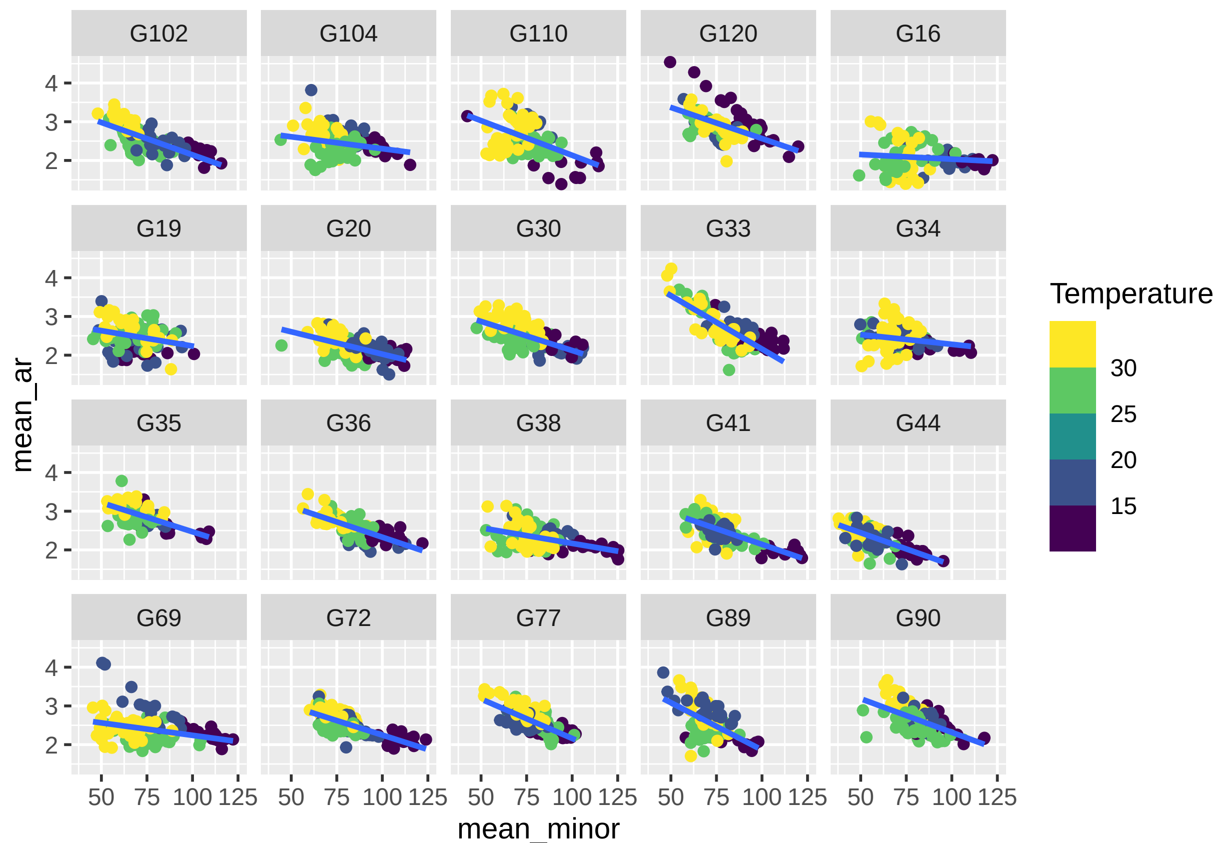


Here, we do see that some outcrossed lines seem to show a relationship between speed and length, but, again, the relationships aren't quite as consistent as the relationships between length and width.

We can also look at the relationships between width, aspect ratio, and speed.

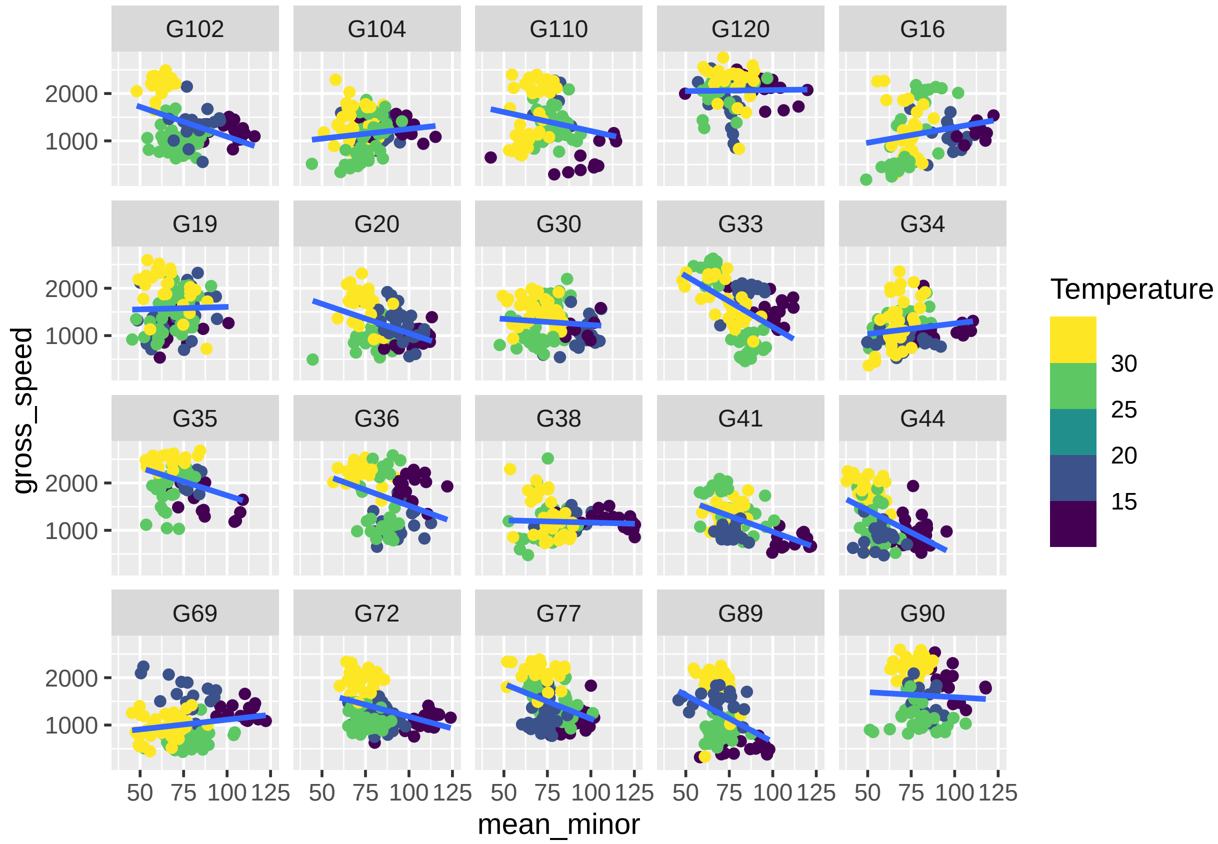
```
ggplot(data = morph_data, aes(x = mean_minor, y = mean_ar)) + geom_point(aes(color = Temperature)) + ge
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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




```
ggplot(data = morph_data, aes(x = mean_minor, y = gross_speed)) + geom_point(aes(color = Temperature)) +
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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

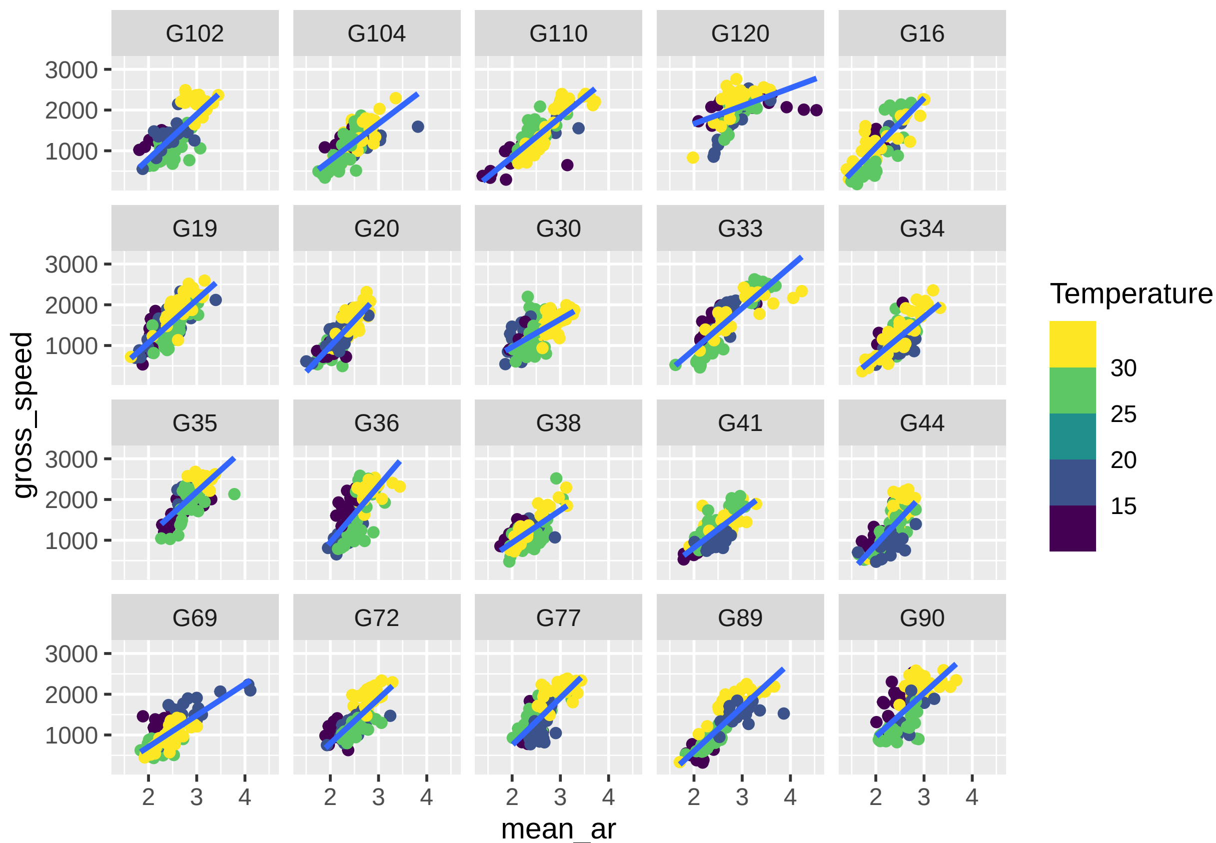


In the width and aspect ratio plots we see a weak negative relationship between width and aspect ratio as we might expect. This appears to translate to negative relationships between width and speed in some outcrossed lines but not others.

Last, we can look at the relationship between aspect ratio and speed.

```
ggplot(data = morph_data, aes(x = mean_ar, y = gross_speed)) + geom_point(aes(color = Temperature)) + ge
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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



Here, again, we see fairly strong and consistent relationships between aspect ratio and speed as we had found previously across outcrossed lines in length and width. There is also a tendency for the highest aspect ratios and speeds for each outcrossed line to be at the higher temperatures.

What does this mean overall? I think that it supports the idea that plasticity is occurring on the same general axes within outcrossed lines as we see for genetic variation across outcrossed lines. That is, there are strong correlations in plasticity between length and width and between aspect ratio and speed, but weaker correlations across the combinations of the variables just as we saw with variation across outcrossed lines at a single temperature.

Quantifying Plasticity

Now that we have looked at how the different outcrossed lines are responding to temperature in their morphology and movement, we want to look at how plastic each of the outcrossed lines are in their phenotypes. To do this, we will first define the degree of plasticity as the range of the mean trait across the temperature range for each outcrossed line. This feels like a fairly intuitive, ad hoc way to define the degree of plasticity.

```
### get data for length

summ_length <- morph_data %>% group_by(Line, Temperature) %>% summarise(length = median(mean_major)) %>%
  summarise(mean_length = median(length),
            max_length = max(length),
            min_length = min(length),
            range_length = max_length - min_length,
            stand_range_length = range_length/max_length)
```

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

```
### predictions for width

summ_width <- morph_data %>% group_by(Line, Temperature) %>% summarise(width = median(mean_minor)) %>% u
  summarise(mean_width = mean(width),
            max_width = max(width),
            min_width = min(width),
            range_width = max_width - min_width,
            stand_range_width = range_width/max_width)
```

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

```
### predictions for aspect ratio

summ_ar <- morph_data %>% group_by(Line, Temperature) %>% summarise(ar = median(mean_ar)) %>% ungroup()
  summarise(mean_ar = mean(ar),
            max_ar = max(ar),
            min_ar = min(ar),
            range_ar = max_ar - min_ar,
            stand_range_ar = range_ar/max_ar)
```

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

```
### predictions for speed

summ_speed <- morph_data %>% group_by(Line, Temperature) %>% summarise(speed = median(gross_speed)) %>%
  summarise(mean_speed = mean(speed),
            max_speed = max(speed),
            min_speed = min(speed),
            range_speed = max_speed - min_speed,
            stand_range_speed = range_speed/max_speed)
```

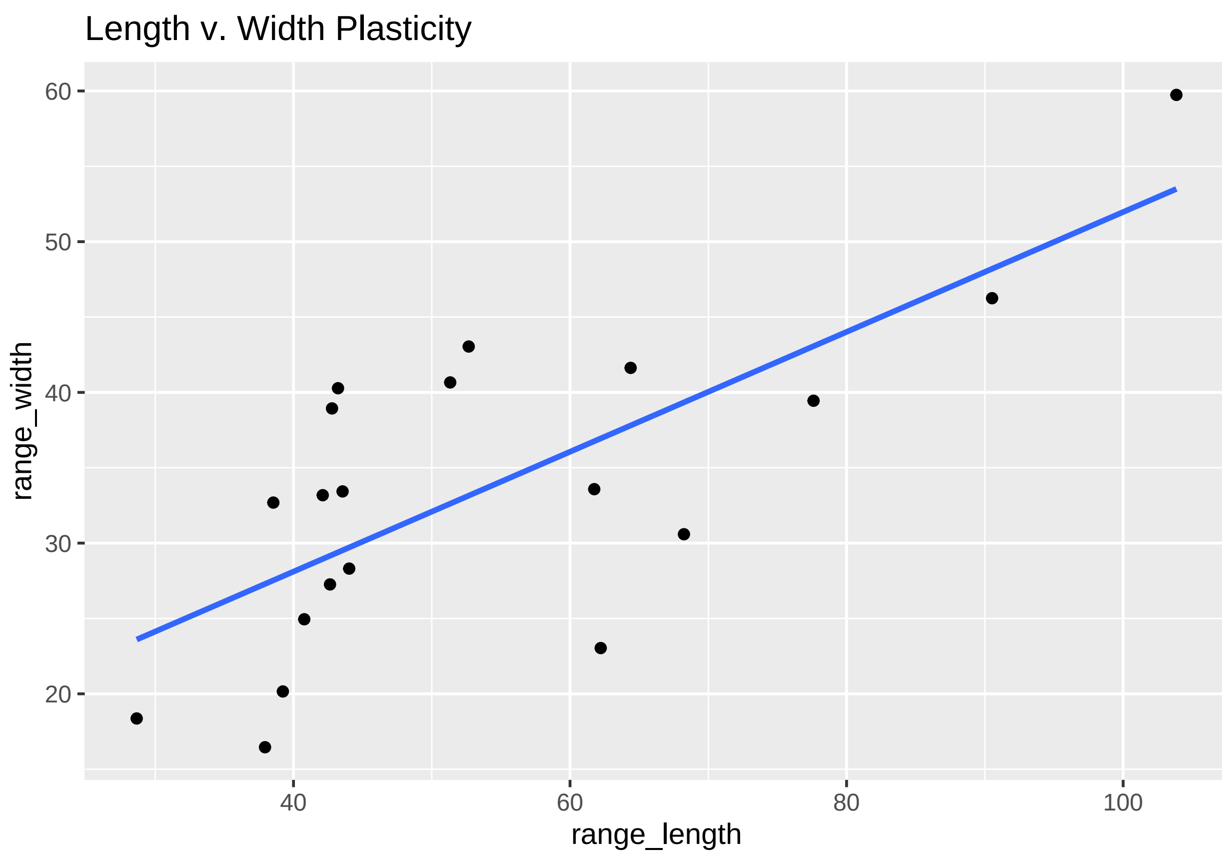
```
## `summarise()` has grouped output by 'Line'. You can override using the
## `.groups` argument.
```

```
### put these together into a single data frame

plast_morph_data <- full_join(summ_length, summ_width, by = "Line")

plast_morph_data <- full_join(plast_morph_data, summ_ar, by = "Line")

plast_morph_data <- full_join(plast_morph_data, summ_speed, by = "Line")
```

Now that we have quantified plasticity in each of the phenotypes, we can ask whether plasticity in certain phenotypes are related to one another. That is, are outcrossed lines that are more plastic in length also more plastic in width, aspect ratio, or speed?

```
### length versus width plasticity

ggplot(data = plast_morph_data, aes(x = range_length, y = range_width)) + geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + ggtitle('Length v. Width Plasticity')
```

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

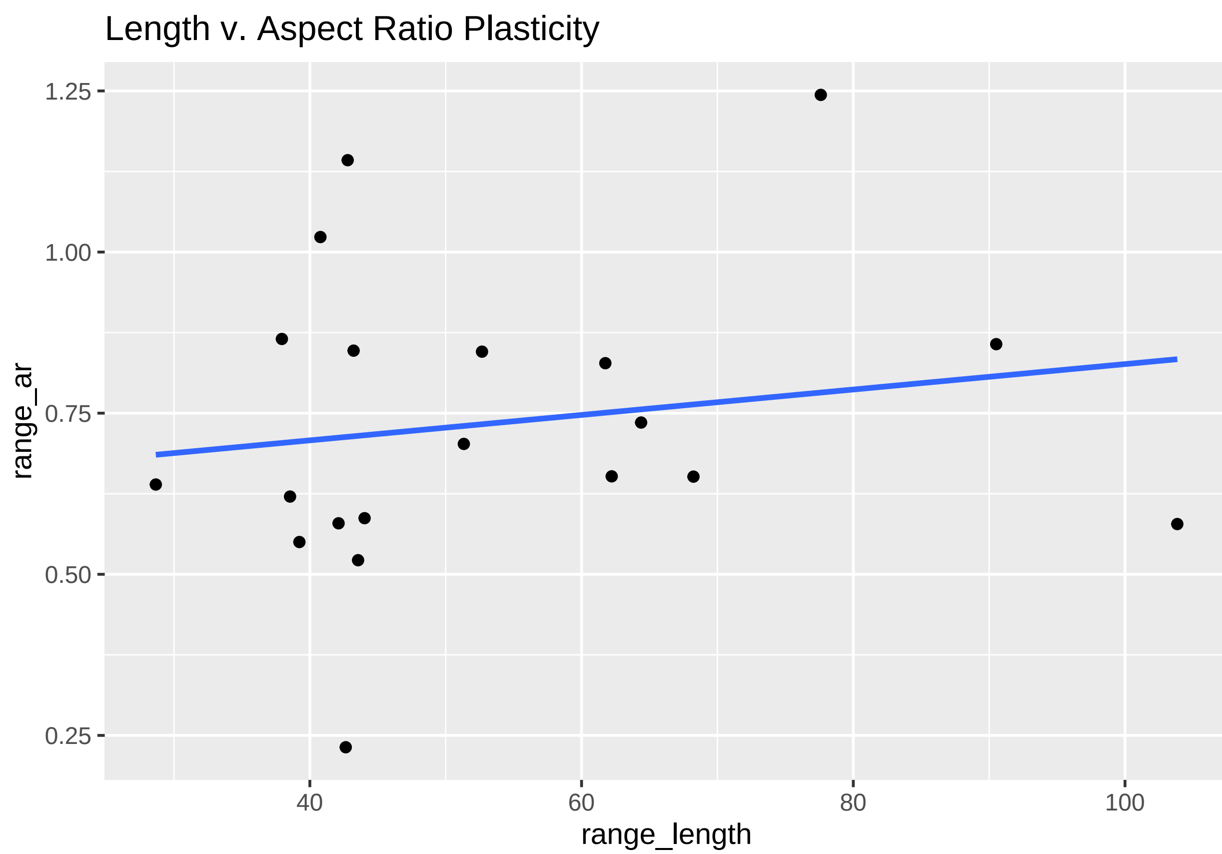



```
summary(lm(range_width ~ range_length, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_width ~ range_length, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.908  -4.024  -1.652   5.429  10.886
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  12.20494    5.06765   2.408 0.026958 *
## range_length  0.39762    0.08891   4.472 0.000295 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.479 on 18 degrees of freedom
## Multiple R-squared:  0.5263, Adjusted R-squared:    0.5
## F-statistic:    20 on 1 and 18 DF,  p-value: 0.0002945
```

```
### length versus ar plasticity

ggplot(data = plast_morph_data, aes(x = range_length, y = range_ar)) + geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + ggtitle('Length v. Aspect Ratio Plasticity')
```

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

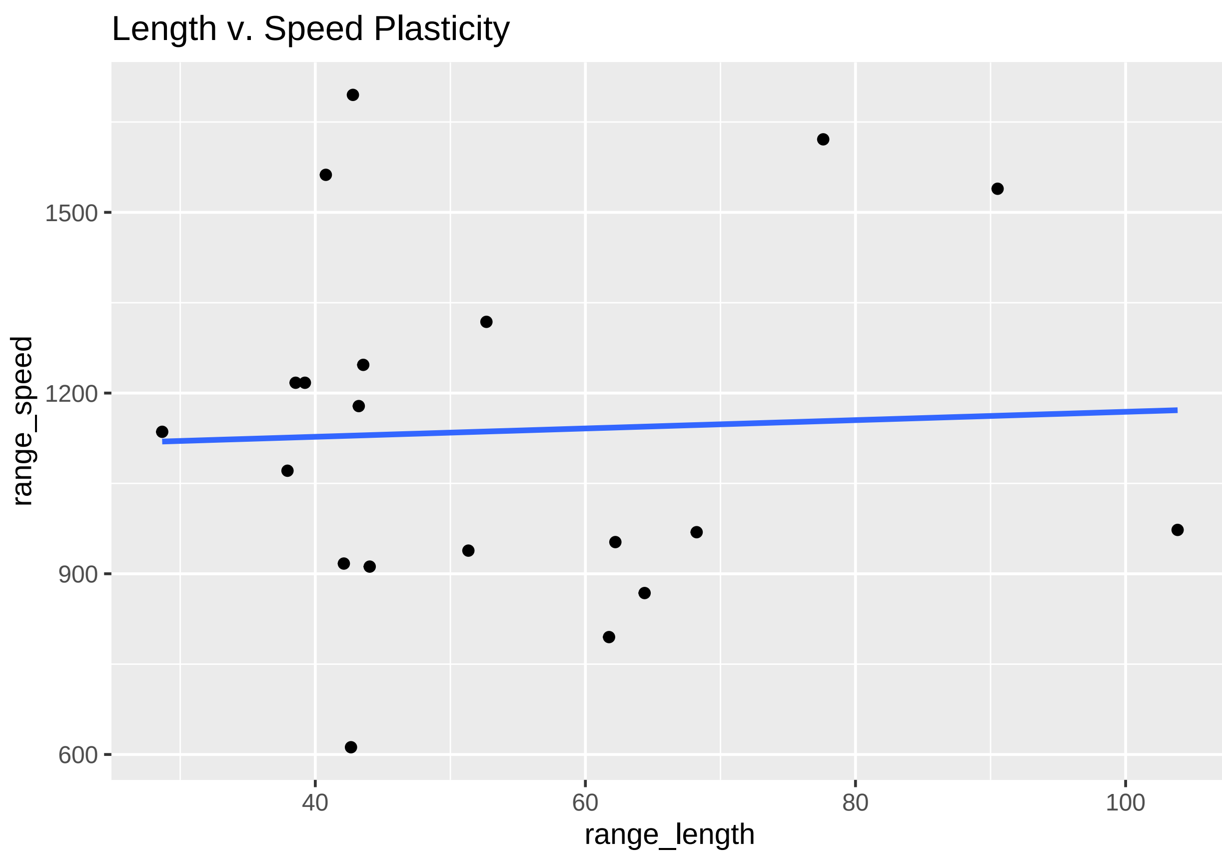



```r
summary(lm(range_ar ~ range_length, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_ar ~ range_length, data = plast_morph_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48139 -0.12975 -0.03703  0.11770  0.46187
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.628943   0.158662   3.964  0.00091 ***
## range_length 0.001972   0.002784   0.708  0.48775
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2341 on 18 degrees of freedom
## Multiple R-squared:  0.02713,    Adjusted R-squared:  -0.02692
## F-statistic: 0.5019 on 1 and 18 DF,  p-value: 0.4878
```

```r
### length versus speed plasticity

ggplot(data = plast_morph_data, aes(x = range_length, y = range_speed)) + geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + ggtitle('Length v. Speed Plasticity')
```

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

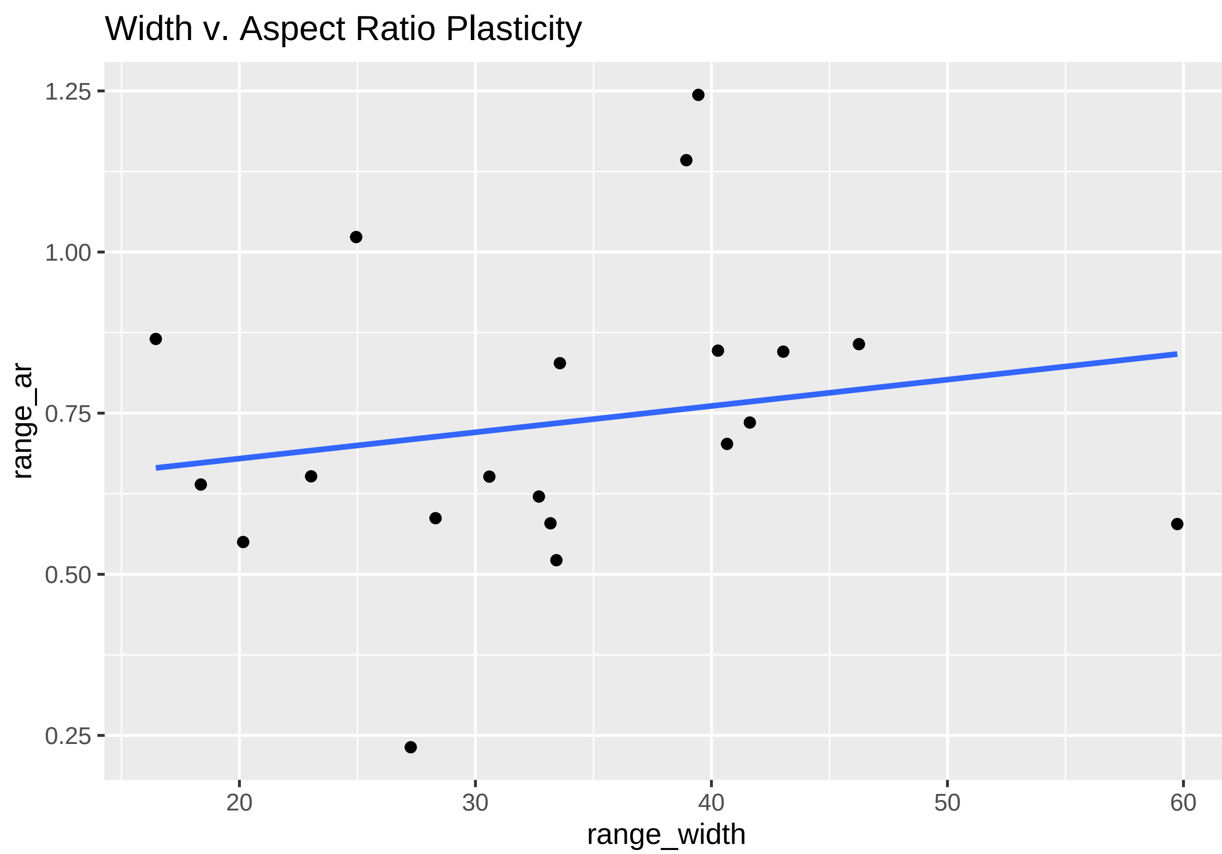



```r
summary(lm(range_speed ~ range_length, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_speed ~ range_length, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -517.19 -202.03  -19.39  133.24  565.66
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1099.7380   204.6302   5.374 4.16e-05 ***
## range_length    0.6915     3.5902   0.193    0.849
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 302 on 18 degrees of freedom
## Multiple R-squared:  0.002057,	Adjusted R-squared:  -0.05338
## F-statistic: 0.0371 on 1 and 18 DF,  p-value: 0.8494
```

```r
### width versus aspect ratio plasticity

ggplot(data = plast_morph_data, aes(x = range_width, y = range_ar)) + geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + ggtitle('Width v. Aspect Ratio Plasticity')
```

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

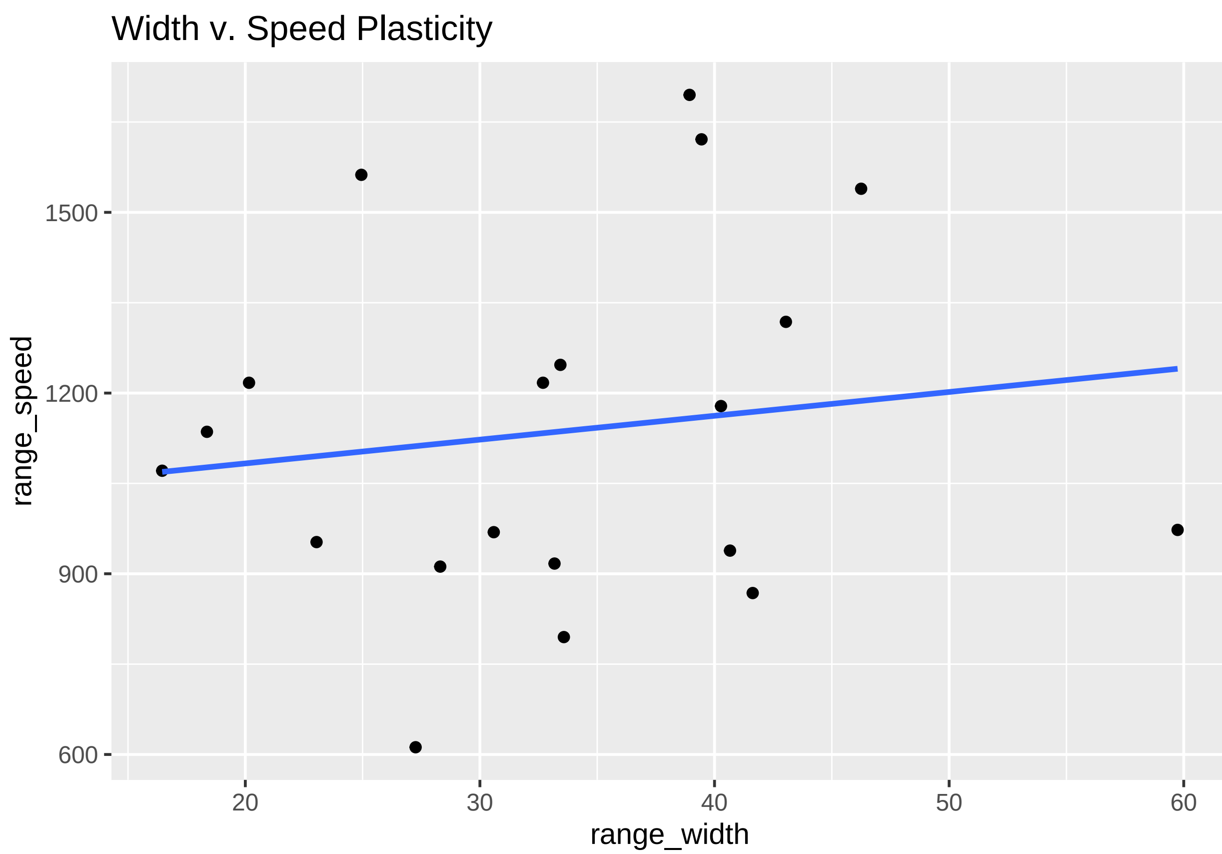



```r
summary(lm(range_ar ~ range_width, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_ar ~ range_width, data = plast_morph_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.47752 -0.12733 -0.03674  0.08670  0.48494
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.597876   0.177781   3.363  0.00346 **
## range_width 0.004083   0.005059   0.807  0.43016
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2332 on 18 degrees of freedom
## Multiple R-squared:  0.03492,    Adjusted R-squared:  -0.01869
## F-statistic: 0.6514 on 1 and 18 DF,  p-value: 0.4302
```

```r
### width versus speed plasticity

ggplot(data = plast_morph_data, aes(x = range_width, y = range_speed)) + geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + ggtitle('Width v. Speed Plasticity')
```

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

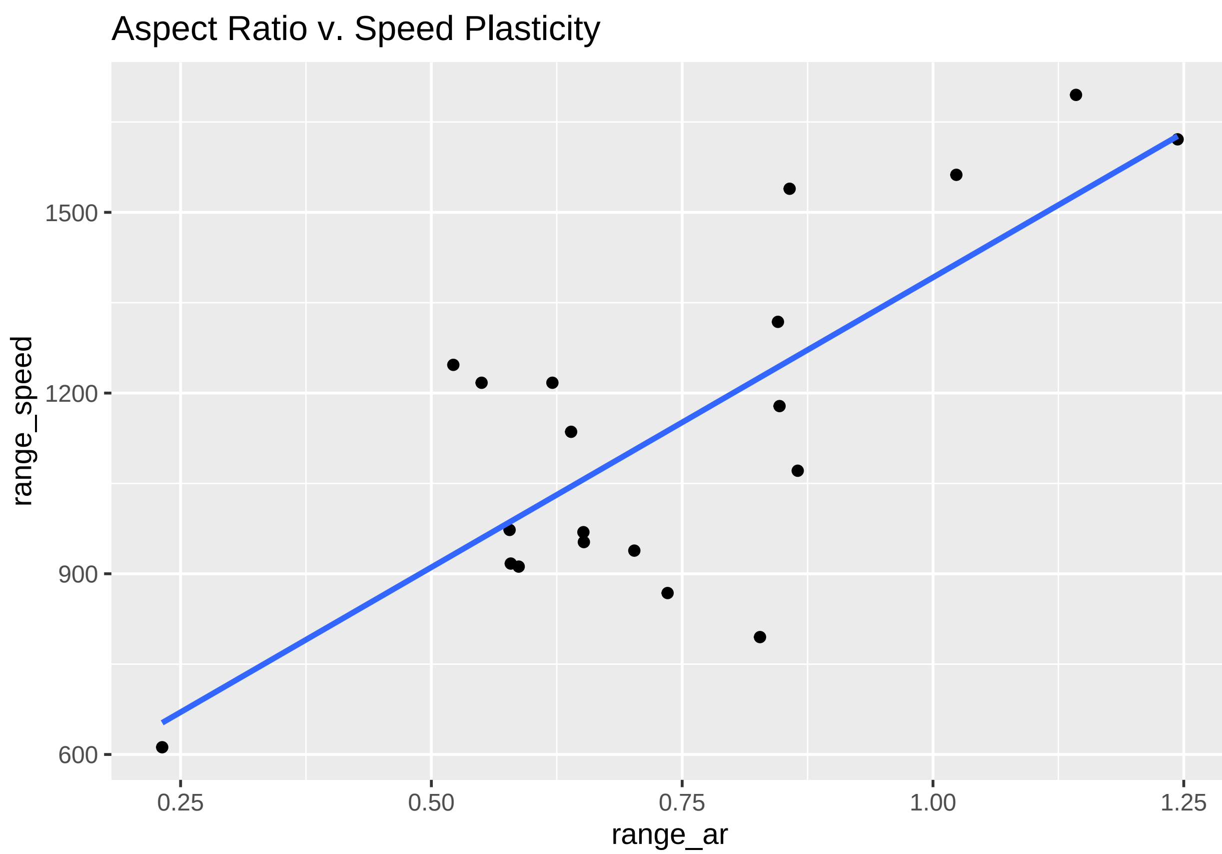



```r
summary(lm(range_speed ~ range_width, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_speed ~ range_width, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -499.83 -220.35    8.48  135.97  536.94
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1004.072    228.109   4.402 0.000344 ***
## range_width    3.955      6.491   0.609 0.549947
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 299.2 on 18 degrees of freedom
## Multiple R-squared:  0.02021,    Adjusted R-squared:  -0.03423
## F-statistic: 0.3712 on 1 and 18 DF,  p-value: 0.5499
```

```r
### aspect ratio versus speed plasticity

ggplot(data= plast_morph_data, aes(x = range_ar, y = range_speed)) + geom_point() +
  geom_smooth(method = 'lm', se = FALSE) + ggtitle('Aspect Ratio v. Speed Plasticity')
```

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




```r
summary(lm(range_speed ~ range_ar, data = plast_morph_data))
```

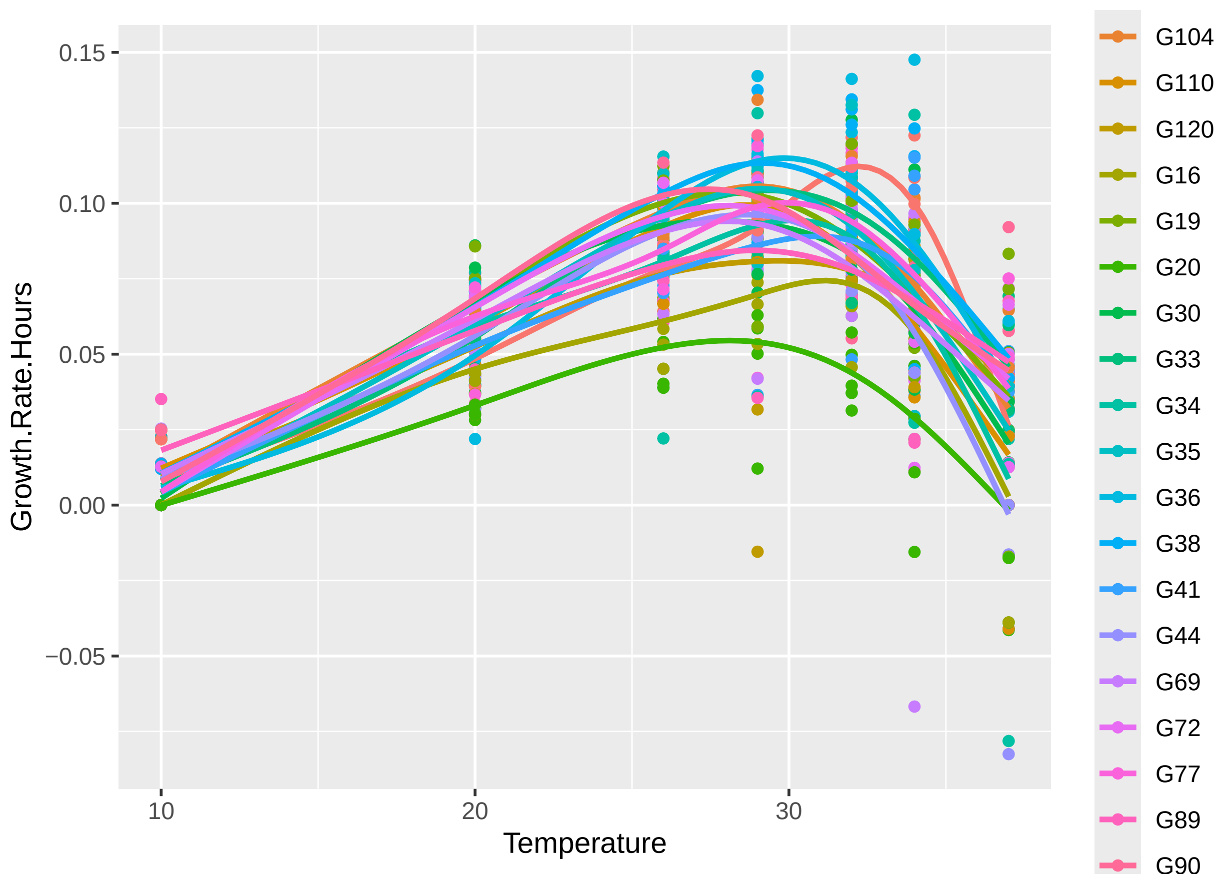
```
##
## Call:
## lm(formula = range_speed ~ range_ar, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -431.06  -91.75  -26.77  152.48  315.00
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    429.7      151.1   2.843 0.010796 *
## range_ar       962.2      196.6   4.894 0.000117 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 198 on 18 degrees of freedom
## Multiple R-squared:  0.5709, Adjusted R-squared:  0.5471
## F-statistic: 23.95 on 1 and 18 DF,  p-value: 0.0001169
```

These regressions show fairly strong evidence in our data that plasticity in length and width are associated with one another as well as plasticity in aspect ratio and speed, but not among any of the variables between these groups. This suggests that outcrossed lines can vary plastically in size while also varying plastically in aspect ratio/speed either a lot or a little. Again, since we know that these axes of variation show some amount of genetic independence, this suggests that the plasticity in these two sets of traits also varies independently.

Growth across temperatures

Now, we want to determine whether there are relationships between the phenotypes of outcrossed lines across temperatures and the population growth rates of the outcrossed lines. First, we will load the growth rate data and make some modifications and then combine the growth rate and phenotype data. To combine the data, we will use the means of the phenotypes as the 'observed' phenotype of the outcrossed line at each temperature.

```
growth_data <- read.csv('StartPop_TPC.csv')

growth_data <- growth_data %>% filter(!is.na(Growth.Rate.Hours) & !is.infinite(Growth.Rate.Hours))

growth_data <- growth_data %>% filter(Genotype != 'blank')

growth_data$Genotype <- as.factor(paste0('G', growth_data$Genotype))

colnames(growth_data)[which(colnames(growth_data) == 'Genotype')] <- 'Line'

### take a look at the growth rate data

ggplot(data = growth_data, aes(x = Temperature, y = Growth.Rate.Hours, color = Line)) +
  geom_point() + geom_smooth(method = 'gam', formula = y ~ s(x, k = 6, bs = 'tp'), se = FALSE)
```




```
### set up a phenotype data frame to join with the growth dataframe

morph_combine <- morph_data %>% group_by(Line, Temperature) %>%
  summarise(mean_length = mean(mean_major),
```

```
            mean_width = mean(mean_minor),
            mean_ar = mean(mean_ar),
            mean_speed = mean(gross_speed))
```

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

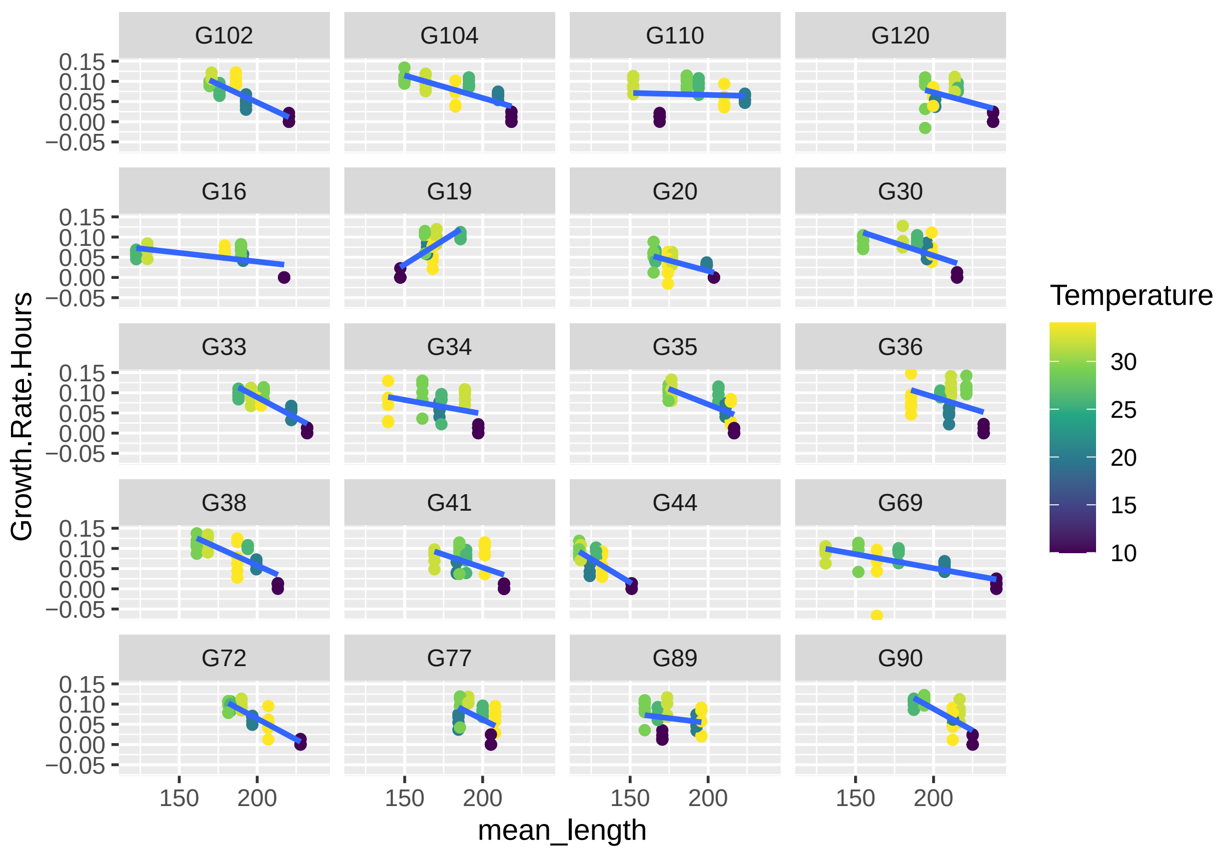
```
growth_combine <- growth_data %>% filter(Temperature <= 34)

growth_phenotypes <- left_join(growth_combine, morph_combine, by = c('Line', 'Temperature'))
```

Length

Now that we have combined the predictions of the morphological and movement data with the growth data across temperatures, we can look and see if there are relationships between plasticity in morphology and movement and growth. We will start by looking at length.

```
ggplot(data = growth_phenotypes, aes(x = mean_length, y = Growth.Rate.Hours)) + geom_point(aes(color = 
  geom_smooth(method = 'lm', se = FALSE) + facet_wrap(facets = "Line", nrow = 5, ncol = 4) +
  scale_color_viridis_c()
```

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




```
fit_length_growth <- lm(Growth.Rate.Hours ~ mean_length*Line, data = growth_phenotypes)

summary(fit_length_growth)
```

```
##
```

```
## Call:
## lm(formula = Growth.Rate.Hours ~ mean_length * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.143057 -0.017333  0.000466  0.019207  0.077231
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            4.086e-01  4.915e-02   8.314 5.10e-16 ***
## mean_length           -1.803e-03  2.632e-04  -6.851 1.66e-11 ***
## LineG104              -1.264e-01  6.084e-02  -2.078 0.038098 *
## LineG110              -3.230e-01  6.128e-02  -5.270 1.84e-07 ***
## LineG120              -1.278e-01  8.658e-02  -1.476 0.140460
## LineG16               -2.827e-01  5.450e-02  -5.188 2.81e-07 ***
## LineG19               -7.398e-01  8.347e-02  -8.863  < 2e-16 ***
## LineG20               -1.844e-01  7.389e-02  -2.495 0.012827 *
## LineG30               -1.015e-01  6.812e-02  -1.490 0.136654
## LineG33                9.004e-02  8.042e-02   1.120 0.263231
## LineG34               -2.232e-01  6.510e-02  -3.429 0.000642 ***
## LineG35               -3.351e-02  7.203e-02  -0.465 0.641977
## LineG36               -8.285e-02  8.370e-02  -0.990 0.322629
## LineG38               -2.252e-03  6.916e-02  -0.033 0.974037
## LineG41               -9.709e-02  7.933e-02  -1.224 0.221428
## LineG44               -4.402e-02  7.199e-02  -0.611 0.541149
## LineG69               -2.189e-01  5.436e-02  -4.027 6.30e-05 ***
## LineG72                6.830e-02  7.532e-02   0.907 0.364839
## LineG77                4.113e-02  1.089e-01   0.378 0.705743
## LineG89               -2.592e-01  7.907e-02  -3.278 0.001098 **
## LineG90                1.067e-01  8.832e-02   1.208 0.227330
## mean_length:LineG104  6.867e-04  3.254e-04   2.110 0.035196 *
## mean_length:LineG110  1.708e-03  3.257e-04   5.243 2.12e-07 ***
## mean_length:LineG120  7.628e-04  4.294e-04   1.777 0.076097 .
## mean_length:LineG16   1.369e-03  2.956e-04   4.632 4.34e-06 ***
## mean_length:LineG19   4.225e-03  4.825e-04   8.757  < 2e-16 ***
## mean_length:LineG20   7.622e-04  4.021e-04   1.896 0.058439 .
## mean_length:LineG30   5.378e-04  3.618e-04   1.486 0.137644
## mean_length:LineG33  -2.456e-04  4.037e-04  -0.609 0.543043
## mean_length:LineG34   1.115e-03  3.607e-04   3.092 0.002070 **
## mean_length:LineG35   2.843e-04  3.715e-04   0.765 0.444415
## mean_length:LineG36   6.232e-04  4.150e-04   1.502 0.133643
## mean_length:LineG38   6.074e-05  3.691e-04   0.165 0.869325
## mean_length:LineG41   5.079e-04  4.192e-04   1.212 0.226043
## mean_length:LineG44  -5.215e-04  4.854e-04  -1.075 0.282965
## mean_length:LineG69   1.110e-03  2.924e-04   3.796 0.000160 ***
## mean_length:LineG72  -2.611e-04  3.900e-04  -0.670 0.503346
## mean_length:LineG77  -1.366e-04  5.615e-04  -0.243 0.807900
## mean_length:LineG89   1.322e-03  4.377e-04   3.021 0.002618 **
## mean_length:LineG90  -3.315e-04  4.396e-04  -0.754 0.451015
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02763 on 676 degrees of freedom
## Multiple R-squared:  0.4636, Adjusted R-squared:  0.4327
```

```
## F-statistic: 14.98 on 39 and 676 DF,  p-value: < 2.2e-16
```

```r
fit_length_growth_noint <- lm(Growth.Rate.Hours ~ mean_length + Line, data = growth_phenotypes)

AIC(fit_length_growth, fit_length_growth_noint)
```

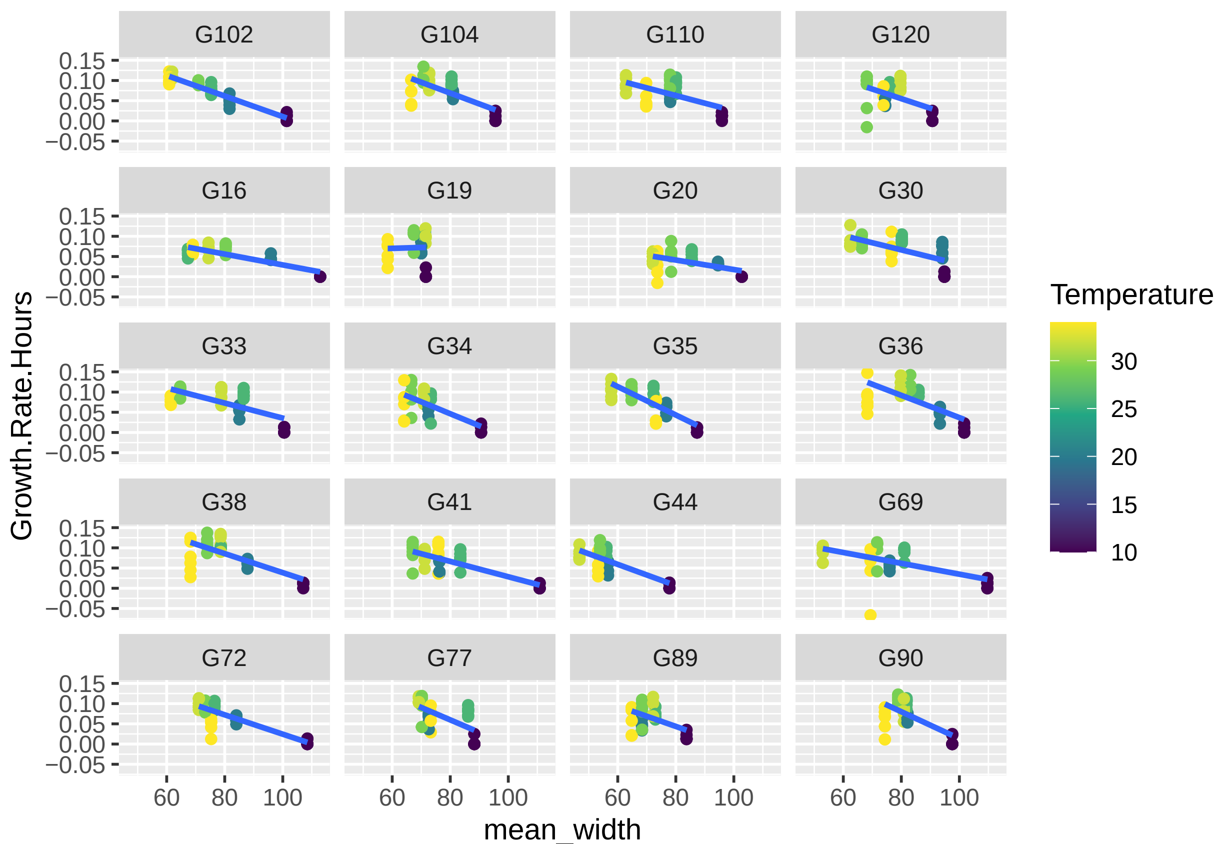
```
##                         df       AIC
## fit_length_growth       41 -3066.234
## fit_length_growth_noint 22 -2923.264
```

In general, we see that there is a negative relationship between length and growth rates across temperatures in each of the genotypes. A statistical model with an interaction between length and and outcrossed line also outperforms a model with no interaction suggesting that there is significant variation among lines in the relationship between body size change and growth rate change. G19 is a bit weird . . .

Width

```r
ggplot(data = growth_phenotypes, aes(x = mean_width, y = Growth.Rate.Hours)) + geom_point(aes(color = Te
  geom_smooth(method = 'lm', se = FALSE) + facet_wrap(facets = "Line", nrow = 5, ncol = 4) +
  scale_color_viridis_c()
```

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




```r
fit_width_growth <- lm(Growth.Rate.Hours ~ mean_width*Line, data = growth_phenotypes)

summary(fit_width_growth)
```

```
##
```

```
## Call:
## lm(formula = Growth.Rate.Hours ~ mean_width * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.142682 -0.014480  0.001527  0.016856  0.058821
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.652e-01  2.472e-02  10.729  < 2e-16 ***
## mean_width          -2.546e-03  3.228e-04  -7.887 1.24e-14 ***
## LineG104             1.770e-02  4.428e-02   0.400  0.68954
## LineG110            -5.083e-02  4.204e-02  -1.209  0.22709
## LineG120            -2.023e-02  5.764e-02  -0.351  0.72571
## LineG16             -1.021e-01  3.386e-02  -3.015  0.00267 **
## LineG19             -2.082e-01  6.949e-02  -2.997  0.00283 **
## LineG20             -1.298e-01  4.193e-02  -3.096  0.00205 **
## LineG30             -5.773e-02  3.780e-02  -1.527  0.12714
## LineG33             -4.343e-02  3.639e-02  -1.193  0.23320
## LineG34              1.779e-02  4.548e-02   0.391  0.69582
## LineG35              5.838e-02  4.262e-02   1.370  0.17119
## LineG36              4.673e-02  4.397e-02   1.063  0.28830
## LineG38              9.009e-03  3.845e-02   0.234  0.81483
## LineG41             -4.667e-02  3.538e-02  -1.319  0.18755
## LineG44             -4.885e-02  3.634e-02  -1.344  0.17931
## LineG69             -9.618e-02  3.203e-02  -3.002  0.00278 **
## LineG72             -1.329e-03  3.782e-02  -0.035  0.97197
## LineG77              4.075e-02  5.094e-02   0.800  0.42395
## LineG89             -2.041e-02  5.859e-02  -0.348  0.72770
## LineG90              8.152e-02  5.673e-02   1.437  0.15119
## mean_width:LineG104 -1.297e-04  5.690e-04  -0.228  0.81969
## mean_width:LineG110  6.511e-04  5.420e-04   1.201  0.23005
## mean_width:LineG120  1.717e-04  7.488e-04   0.229  0.81874
## mean_width:LineG16   1.206e-03  4.225e-04   2.855  0.00444 **
## mean_width:LineG19   2.766e-03  1.001e-03   2.763  0.00588 **
## mean_width:LineG20   1.369e-03  5.120e-04   2.673  0.00770 **
## mean_width:LineG30   7.818e-04  4.814e-04   1.624  0.10487
## mean_width:LineG33   6.835e-04  4.626e-04   1.478  0.13998
## mean_width:LineG34  -4.121e-04  6.115e-04  -0.674  0.50057
## mean_width:LineG35  -9.580e-04  5.770e-04  -1.660  0.09730 .
## mean_width:LineG36  -2.055e-04  5.321e-04  -0.386  0.69949
## mean_width:LineG38   1.896e-04  4.787e-04   0.396  0.69211
## mean_width:LineG41   6.450e-04  4.465e-04   1.445  0.14904
## mean_width:LineG44  -7.509e-05  5.600e-04  -0.134  0.89338
## mean_width:LineG69   1.204e-03  4.139e-04   2.909  0.00375 **
## mean_width:LineG72   1.514e-04  4.741e-04   0.319  0.74947
## mean_width:LineG77  -5.315e-04  6.624e-04  -0.802  0.42256
## mean_width:LineG89   3.143e-05  8.055e-04   0.039  0.96889
## mean_width:LineG90  -7.882e-04  6.953e-04  -1.134  0.25733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02654 on 676 degrees of freedom
## Multiple R-squared:  0.5053, Adjusted R-squared:  0.4767
```

```
## F-statistic:  17.7 on 39 and 676 DF,  p-value: < 2.2e-16
```

```
fit_width_growth_noint <- lm(Growth.Rate.Hours ~ mean_width + Line, data = growth_phenotypes)

AIC(fit_width_growth, fit_width_growth_noint)
```

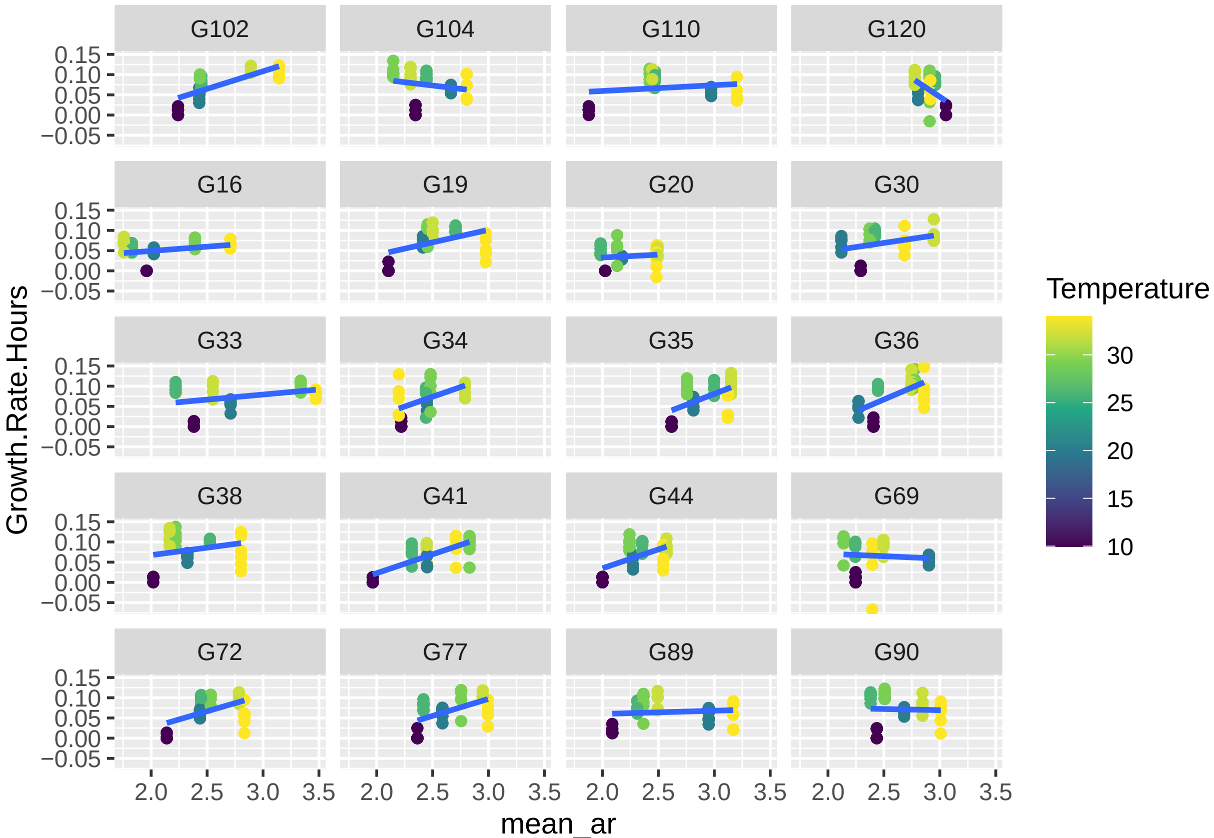
```
##                        df       AIC
## fit_width_growth       41 -3124.107
## fit_width_growth_noint 22 -3104.452
```

As for length, we also see that shorter paramecia within the outcrossed lines have higher growth rates. Again, a model with an interaction between width and outcrossed line is preferred over a model without an interaction.

Aspect Ratio

```
ggplot(data = growth_phenotypes, aes(x = mean_ar, y = Growth.Rate.Hours)) + geom_point(aes(color = Tempe
  geom_smooth(method = 'lm', se = FALSE) + facet_wrap(facets = "Line", nrow = 5, ncol = 4) +
  scale_color_viridis_c()
```

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




```
fit_ar_growth <- lm(Growth.Rate.Hours ~ mean_ar*Line, data = growth_phenotypes)

summary(fit_ar_growth)
```

```
##
## Call:
```

```
## lm(formula = Growth.Rate.Hours ~ mean_ar * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.132852 -0.021274  0.005129  0.024764  0.084685
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.1524428  0.0454944  -3.351 0.000851 ***
## mean_ar           0.0868957  0.0173799   5.000 7.32e-07 ***
## LineG104          0.3080617  0.0756239   4.074 5.18e-05 ***
## LineG110          0.1833447  0.0563352   3.255 0.001192 **
## LineG120          0.7497242  0.1915896   3.913 0.000100 ***
## LineG16           0.1590498  0.0572614   2.778 0.005628 **
## LineG19           0.0670927  0.0688123   0.975 0.329904
## LineG20           0.1577055  0.0750098   2.102 0.035882 *
## LineG30           0.1189353  0.0676157   1.759 0.079032 .
## LineG33           0.1563627  0.0560236   2.791 0.005403 **
## LineG34          -0.0146760  0.0814574  -0.180 0.857075
## LineG35          -0.0879637  0.0932332  -0.943 0.345772
## LineG36          -0.0769907  0.0790224  -0.974 0.330261
## LineG38           0.1461210  0.0674239   2.167 0.030568 *
## LineG41          -0.0115116  0.0662733  -0.174 0.862154
## LineG44           0.0006989  0.0799640   0.009 0.993029
## LineG69           0.2478686  0.0694628   3.568 0.000385 ***
## LineG72           0.0174477  0.0747218   0.234 0.815443
## LineG77          -0.0028039  0.0755019  -0.037 0.970387
## LineG89           0.1960985  0.0589206   3.328 0.000922 ***
## LineG90           0.2389042  0.0785064   3.043 0.002432 **
## mean_ar:LineG104 -0.1199309  0.0300748  -3.988 7.40e-05 ***
## mean_ar:LineG110 -0.0725821  0.0215709  -3.365 0.000809 ***
## mean_ar:LineG120 -0.2710109  0.0663899  -4.082 5.00e-05 ***
## mean_ar:LineG16  -0.0656363  0.0238025  -2.758 0.005981 **
## mean_ar:LineG19  -0.0245855  0.0267474  -0.919 0.358332
## mean_ar:LineG20  -0.0730425  0.0319457  -2.286 0.022536 *
## mean_ar:LineG30  -0.0458639  0.0265828  -1.725 0.084927 .
## mean_ar:LineG33  -0.0617704  0.0208970  -2.956 0.003226 **
## mean_ar:LineG34   0.0094994  0.0327233   0.290 0.771680
## mean_ar:LineG35   0.0202179  0.0328741   0.615 0.538755
## mean_ar:LineG36   0.0317202  0.0303755   1.044 0.296734
## mean_ar:LineG38  -0.0500118  0.0273454  -1.829 0.067856 .
## mean_ar:LineG41   0.0064805  0.0261437   0.248 0.804302
## mean_ar:LineG44   0.0064249  0.0330423   0.194 0.845885
## mean_ar:LineG69  -0.0991354  0.0278164  -3.564 0.000391 ***
## mean_ar:LineG72  -0.0062965  0.0290967  -0.216 0.828741
## mean_ar:LineG77  -0.0027009  0.0283634  -0.095 0.924164
## mean_ar:LineG89  -0.0787903  0.0226021  -3.486 0.000522 ***
## mean_ar:LineG90  -0.0926214  0.0297317  -3.115 0.001916 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03263 on 676 degrees of freedom
## Multiple R-squared:  0.2522, Adjusted R-squared:  0.2091
## F-statistic: 5.847 on 39 and 676 DF,  p-value: < 2.2e-16
```

```r
fit_ar_growth_noint <- lm(Growth.Rate.Hours ~ mean_ar + Line, data = growth_phenotypes)

AIC(fit_ar_growth, fit_ar_growth_noint)
```

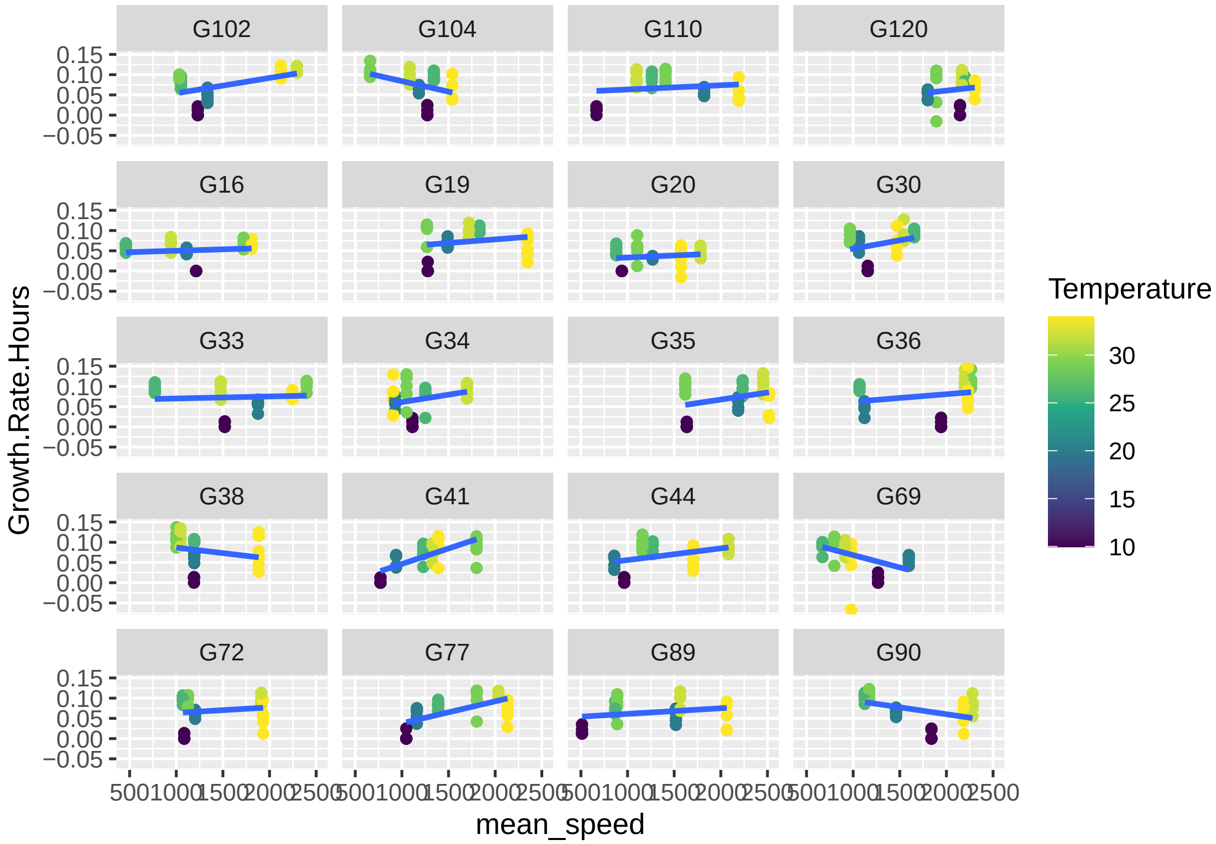
```
##                     df       AIC
## fit_ar_growth       41 -2828.369
## fit_ar_growth_noint 22 -2777.197
```

Although maybe not quite as strong as the length or width relationships, it does appear that paramecia within each outcrossed line generally have higher growth rates when they have higher aspect ratios. Again, a model with an interaction between aspect ratio and outcrossed line performs better than a model without an interaction.

Speed

```r
ggplot(data = growth_phenotypes, aes(x = mean_speed, y = Growth.Rate.Hours)) + geom_point(aes(color = Te
  geom_smooth(method = 'lm', se = FALSE) + facet_wrap(facets = "Line", nrow = 5, ncol = 4) +
  scale_color_viridis_c()
```

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




```r
fit_speed_growth <- lm(Growth.Rate.Hours ~ mean_speed*Line, data = growth_phenotypes)

summary(fit_speed_growth)
```

```
##
## Call:
```

```
## lm(formula = Growth.Rate.Hours ~ mean_speed * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.136344 -0.019439  0.006924  0.024504  0.071491
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.611e-02  1.789e-02   0.901 0.368123
## mean_speed           3.791e-05  1.123e-05   3.376 0.000777 ***
## LineG104             1.201e-01  3.095e-02   3.882 0.000114 ***
## LineG110             3.682e-02  2.482e-02   1.483 0.138440
## LineG120            -6.435e-03  7.246e-02  -0.089 0.929267
## LineG16              2.687e-02  2.401e-02   1.119 0.263469
## LineG19              2.657e-02  3.144e-02   0.845 0.398478
## LineG20              7.054e-03  2.877e-02   0.245 0.806344
## LineG30             -2.510e-03  3.430e-02  -0.073 0.941677
## LineG33              4.910e-02  2.599e-02   1.889 0.059265 .
## LineG34              8.415e-03  3.090e-02   0.272 0.785470
## LineG35             -1.825e-02  3.829e-02  -0.477 0.633693
## LineG36              2.738e-02  2.742e-02   0.998 0.318400
## LineG38              9.828e-02  3.065e-02   3.206 0.001409 **
## LineG41             -4.623e-02  2.842e-02  -1.627 0.104251
## LineG44              1.108e-02  2.588e-02   0.428 0.668724
## LineG69              1.133e-01  2.674e-02   4.239 2.55e-05 ***
## LineG72              3.398e-02  2.798e-02   1.214 0.224983
## LineG77             -3.242e-02  2.859e-02  -1.134 0.257171
## LineG89              3.113e-02  2.305e-02   1.351 0.177264
## LineG90              1.125e-01  2.802e-02   4.013 6.66e-05 ***
## mean_speed:LineG104 -9.012e-05  2.368e-05  -3.805 0.000154 ***
## mean_speed:LineG110 -2.747e-05  1.610e-05  -1.707 0.088351 .
## mean_speed:LineG120 -1.245e-05  3.539e-05  -0.352 0.725142
## mean_speed:LineG16  -3.087e-05  1.671e-05  -1.847 0.065202 .
## mean_speed:LineG19  -2.020e-05  1.893e-05  -1.067 0.286266
## mean_speed:LineG20  -2.774e-05  2.065e-05  -1.344 0.179561
## mean_speed:LineG30   3.609e-06  2.465e-05   0.146 0.883617
## mean_speed:LineG33  -3.294e-05  1.536e-05  -2.144 0.032357 *
## mean_speed:LineG34  -1.208e-06  2.399e-05  -0.050 0.959835
## mean_speed:LineG35  -3.077e-06  1.941e-05  -0.159 0.874070
## mean_speed:LineG36  -1.927e-05  1.578e-05  -1.222 0.222256
## mean_speed:LineG38  -6.532e-05  2.239e-05  -2.917 0.003649 **
## mean_speed:LineG41   3.862e-05  2.052e-05   1.882 0.060201 .
## mean_speed:LineG44  -9.036e-06  1.740e-05  -0.519 0.603688
## mean_speed:LineG69  -9.905e-05  2.151e-05  -4.605 4.92e-06 ***
## mean_speed:LineG72  -2.438e-05  1.870e-05  -1.304 0.192748
## mean_speed:LineG77   1.633e-05  1.758e-05   0.929 0.353234
## mean_speed:LineG89  -2.400e-05  1.559e-05  -1.539 0.124248
## mean_speed:LineG90  -7.205e-05  1.674e-05  -4.305 1.92e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03408 on 676 degrees of freedom
## Multiple R-squared:  0.1844, Adjusted R-squared:  0.1374
## F-statistic:  3.92 on 39 and 676 DF,  p-value: 1.513e-13
```

```r
fit_speed_growth_noint <- lm(Growth.Rate.Hours ~ mean_speed + Line, data = growth_phenotypes)

fit_speed_growth_line <- lm(Growth.Rate.Hours ~ Line, data = growth_phenotypes)

fit_speed_growth_null <- lm(Growth.Rate.Hours ~ 1, data = growth_phenotypes)

AIC(fit_speed_growth, fit_speed_growth_noint, fit_speed_growth_line, fit_speed_growth_null)
```

```
##                        df       AIC
## fit_speed_growth       41 -2766.203
## fit_speed_growth_noint 22 -2723.756
## fit_speed_growth_line  21 -2708.006
## fit_speed_growth_null   2 -2698.239
```

Although the relationships do not appear as consistent between speed and growth, a linear model with an interaction between speed and outcrossed line is the best performing model even when considering a null model with just an intercept and model that only includes line and does not include speed at all.

Within line varation and growth

So, what does all of this mean? Well, it appears that within outcrossed lines, changes in size, aspect ratio, and speed are associated with changes in growth rates. In general, smaller, thinner, faster paramecium have higher growth rates.

Plasticity in morphology and movement

Next, we want to ask whether the extent to which an outcrossed line is plastic in its morphology and movement is related to the extent to which its growth rate changes across temperatures. First, we will just examine plasticity in each of the phenotypes and whether it is associated with the extent to which growth changed for each outcrossed line. We will quantify plasticity in growth similar to the way that we quantified plasticity in morphology and movement. That is, by averaging the measured growth rates at each of the temperatures and then calculating the range in growth across temperatures. Before, we do that, we will fit a GAM to the TPC data just to examine differences among outcrossed lines.

```r
### fit a gam to the growth data

tpc_fit <- gam(formula = Growth.Rate.Hours ~ Line + s(Temperature, k = 5, bs = 'tp', by = Line), data =

summary(tpc_fit)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## Growth.Rate.Hours ~ Line + s(Temperature, k = 5, bs = "tp", by = Line)
##
## Parametric coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0665704  0.0030472  21.846  < 2e-16 ***
## LineG104     0.0019062  0.0043094   0.442   0.6584
## LineG110    -0.0060922  0.0043094  -1.414   0.1579
## LineG120    -0.0087533  0.0044548  -1.965   0.0498 *
## LineG16     -0.0222880  0.0043094  -5.172 2.98e-07 ***
## LineG19      0.0003257  0.0043094   0.076   0.9398
```

```
## LineG20     -0.0364423  0.0043383  -8.400 2.26e-16 ***
## LineG30     -0.0058342  0.0043094  -1.354   0.1762
## LineG33      0.0030447  0.0043094   0.707   0.4801
## LineG34     -0.0085570  0.0043094  -1.986   0.0474 *
## LineG35     -0.0018822  0.0043094  -0.437   0.6624
## LineG36      0.0050241  0.0043094   1.166   0.2441
## LineG38      0.0091135  0.0043385   2.101   0.0360 *
## LineG41     -0.0062299  0.0043094  -1.446   0.1487
## LineG44     -0.0108752  0.0043094  -2.524   0.0118 *
## LineG69     -0.0048623  0.0043094  -1.128   0.2596
## LineG72     -0.0011178  0.0043094  -0.259   0.7954
## LineG77     -0.0008272  0.0043094  -0.192   0.8478
## LineG89     -0.0047258  0.0043094  -1.097   0.2732
## LineG90      0.0012636  0.0043094   0.293   0.7694
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                           edf Ref.df     F p-value
## s(Temperature):LineG102 3.870  3.989 36.02  <2e-16 ***
## s(Temperature):LineG104 3.590  3.895 30.81  <2e-16 ***
## s(Temperature):LineG110 3.611  3.905 29.38  <2e-16 ***
## s(Temperature):LineG120 3.377  3.771 15.87  <2e-16 ***
## s(Temperature):LineG16  3.732  3.953 22.25  <2e-16 ***
## s(Temperature):LineG19  3.493  3.843 32.13  <2e-16 ***
## s(Temperature):LineG20  3.348  3.752 13.29  <2e-16 ***
## s(Temperature):LineG30  3.575  3.888 31.54  <2e-16 ***
## s(Temperature):LineG33  3.540  3.869 30.22  <2e-16 ***
## s(Temperature):LineG34  3.823  3.979 33.00  <2e-16 ***
## s(Temperature):LineG35  3.663  3.928 35.35  <2e-16 ***
## s(Temperature):LineG36  3.732  3.953 39.66  <2e-16 ***
## s(Temperature):LineG38  3.567  3.884 32.50  <2e-16 ***
## s(Temperature):LineG41  3.622  3.910 23.37  <2e-16 ***
## s(Temperature):LineG44  3.749  3.959 37.67  <2e-16 ***
## s(Temperature):LineG69  3.446  3.816 23.62  <2e-16 ***
## s(Temperature):LineG72  3.378  3.772 27.36  <2e-16 ***
## s(Temperature):LineG77  3.617  3.908 28.25  <2e-16 ***
## s(Temperature):LineG89  3.188  3.632 14.61  <2e-16 ***
## s(Temperature):LineG90  3.450  3.818 27.19  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.736   Deviance explained = 76.5%
## -REML = -1830.3  Scale est. = 0.00038997  n = 833
```

```
tpc_fit_noint <- gam(formula = Growth.Rate.Hours ~ Line + s(Temperature, k = 5, bs = 'tp'), data = grow

AIC(tpc_fit, tpc_fit_noint)
```

```
##                     df       AIC
## tpc_fit       96.71319 -4077.992
## tpc_fit_noint 24.98903 -4083.363
```

The GAMs here actually suggest that a model including Line but no interaction between temperature and Line actually performs best with the growth data. This suggests that there is potentially a genetic effect (moving the line up and down), an environmental effect (how the line changes with temperature), but no gene by environment interaction in the TPCs. That being said, we can still get the plasticity of the growth rate for each of the outcrossed lines.

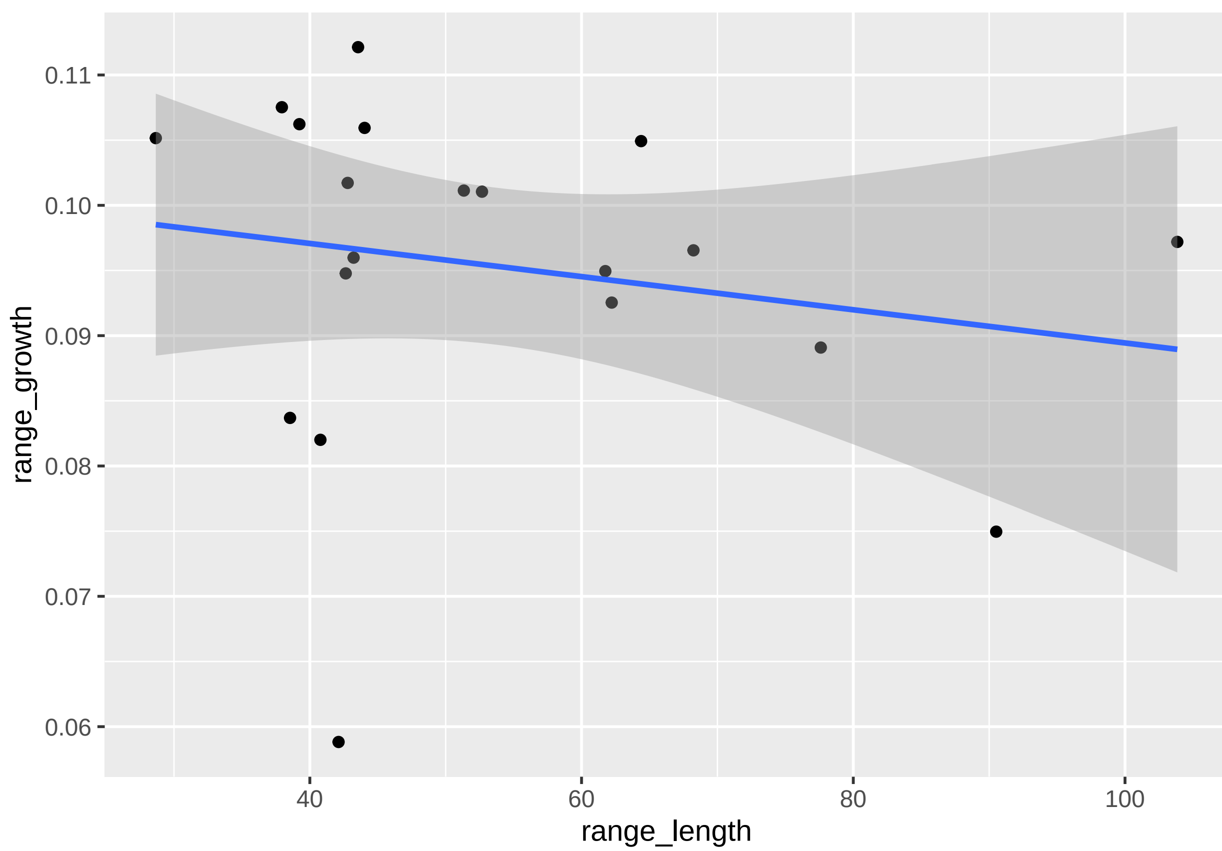
```
tpc_means <- growth_data %>% filter(Temperature <= 34) %>% group_by(Line, Temperature) %>%
  summarise(growth = median(Growth.Rate.Hours)) %>%
  ungroup() %>% group_by(Line) %>%
  summarise(mean_growth = mean(growth),
            max_growth = max(growth),
            min_growth = min(growth),
            range_growth = max_growth - min_growth,
            stand_range_growth = range_growth/max_growth)
```

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

Using the range in growth for the outcrossed lines as our metric of plasticity, we can ask whether the extent of plasticity in any of the morphology or movement variables is predictive of the extent of plasticity in growth rates.

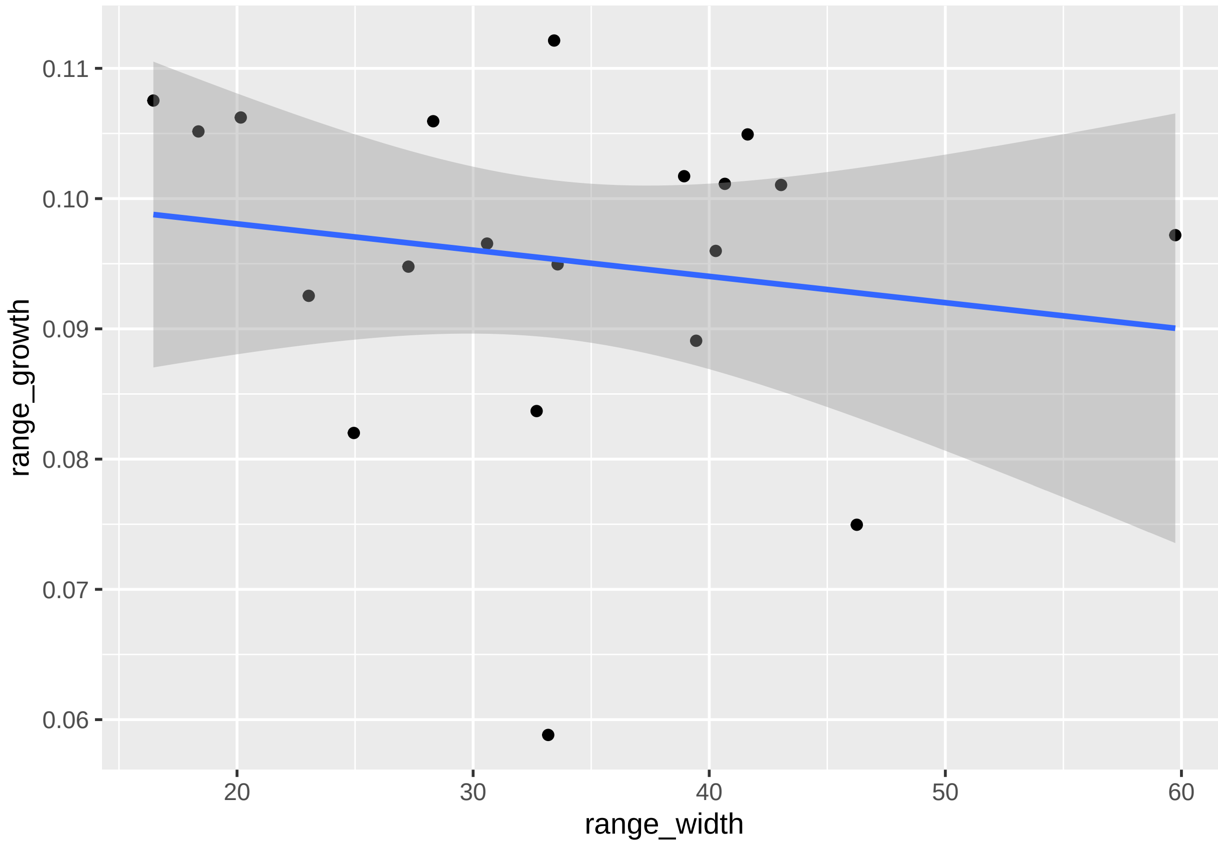
```
plast_growth_data <- full_join(plast_morph_data, tpc_means, by = 'Line')

### length plasticity

ggplot(data = plast_growth_data, aes(x = range_length, y = range_growth)) +
  geom_point() + geom_smooth(method = 'lm', se = TRUE)
```

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



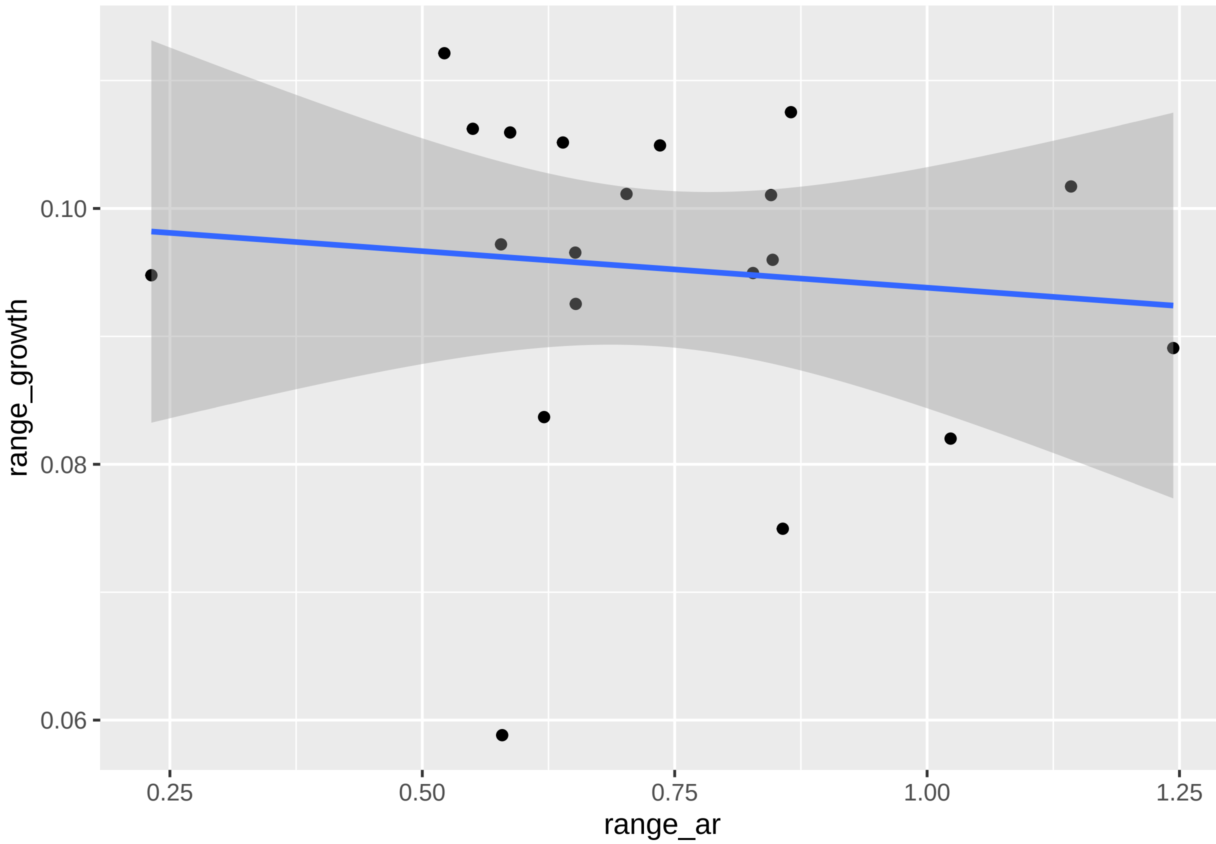
```r
### width plasticity

ggplot(data = plast_growth_data, aes(x = range_width, y = range_growth)) +
  geom_point() + geom_smooth(method = 'lm', se = TRUE)
```

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



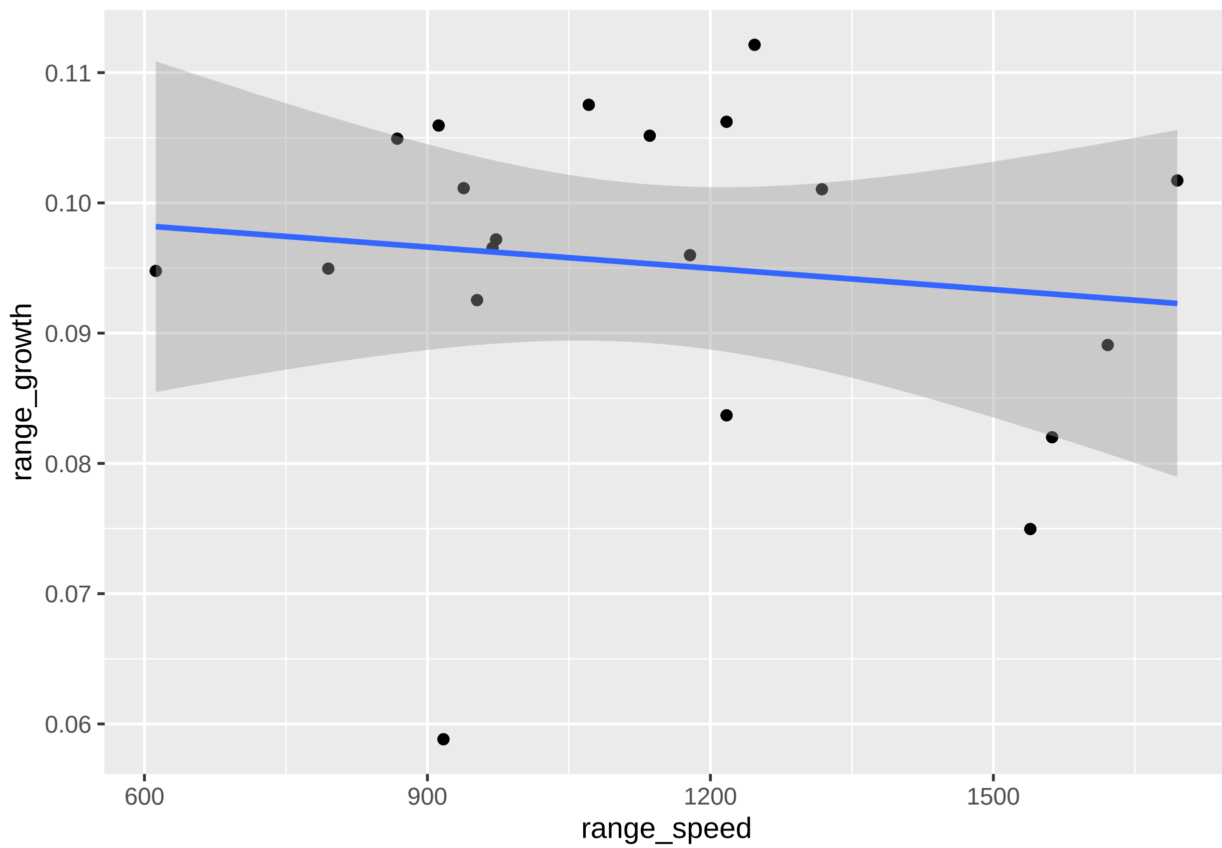

```r
ggplot(data = plast_growth_data, aes(x = range_ar, y = range_growth)) +
  geom_point() + geom_smooth(method = 'lm', se = TRUE)
```

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




```
ggplot(data = plast_growth_data, aes(x = range_speed, y = range_growth)) +
  geom_point() + geom_smooth(method = 'lm', se = TRUE)
```

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



As can be seen from these graphs, there doesn't appear to be any relationship between the extent of phenotypic plasticity in any of the morphological variables and the extent of plasticity in growth. However, as we saw before, the plasticity in size (length/width) is not necessary related to the plasticity in shape and speed. It is possible that we need a measure of multivariate plasticity to compare to the amount of plasticity in growth.

To quantify the amount of multivariate plasticity, we will use a PCA of the morphology data using the means for each line at each temperature. We will then use the area of a convex hull containing those points to be our multivariate measure of plasticity.

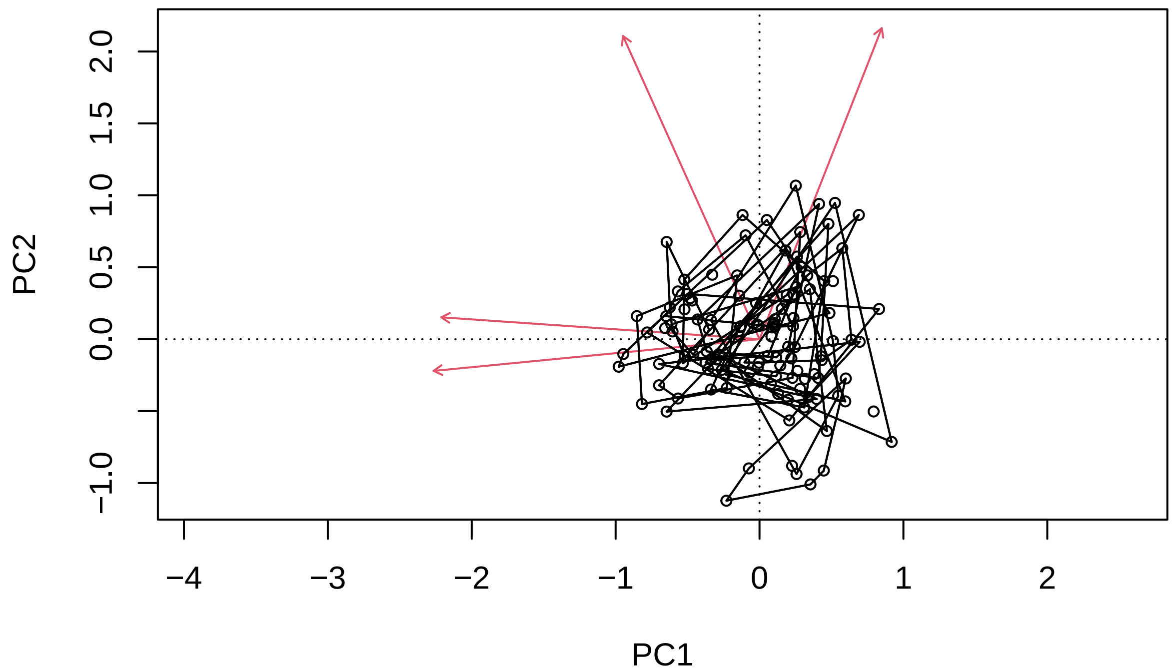
First, we will make perform the PCA.

```r
# perform pca using the rda function in vegan

pca2 <- pca(morph_combine[,3:6], scale = TRUE)

summary(pca2)
```

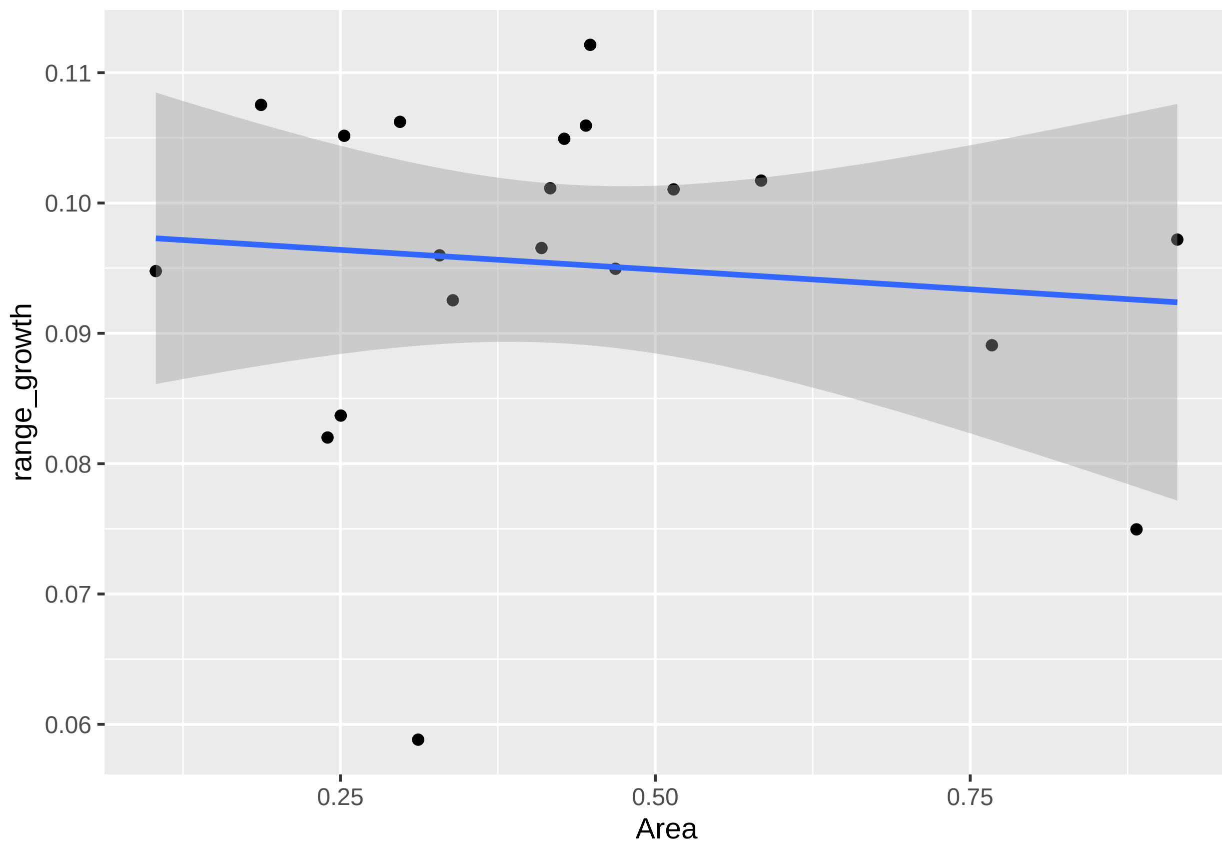
```
##
## Call:
## pca(X = morph_combine[, 3:6], scale = TRUE)
##
## Partitioning of correlations:
##               Inertia Proportion
## Total               4          1
## Unconstrained       4          1
##
## Eigenvalues, and their contribution to the correlations
```

```
##
## Importance of components:
##                           PC1    PC2     PC3      PC4
## Eigenvalue             2.1349 1.6849 0.17207 0.008132
## Proportion Explained   0.5337 0.4212 0.04302 0.002033
## Cumulative Proportion 0.5337 0.9550 0.99797 1.000000
```

```
biplot(pca2)

ordihull(pca2, groups = morph_combine$Line)

hull_data <- t(summary(ordihull(pca2, groups = morph_combine$Line)))
```




```
hull_data <- cbind(rownames(hull_data), data.frame(hull_data, row.names=NULL))

colnames(hull_data)[1] <- 'Line'

plast_growth_data <- full_join(plast_growth_data, hull_data, by = 'Line')

ggplot(data = plast_growth_data, aes(x = Area, y = range_growth)) + geom_point() + geom_smooth(method =

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




```r
summary(lm(range_growth ~ Area, data = plast_growth_data))
```

```
##
## Call:
## lm(formula = range_growth ~ Area, data = plast_growth_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.037206 -0.003542  0.002959  0.008977  0.016931
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.097917   0.006554  14.940 1.38e-11 ***
## Area        -0.006050   0.013681  -0.442    0.664
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01299 on 18 degrees of freedom
## Multiple R-squared:  0.01075,    Adjusted R-squared:  -0.04421
## F-statistic: 0.1956 on 1 and 18 DF,  p-value: 0.6636
```

I think our original hypothesis here was that morphological and movement plasticity would potentially allow the paramecia to buffer the effects of temperature and that this would lead to less plasticity in growth with temperature. The data, however, seem to show that there is no relationship between the degree of change in morphology and movement and the degree of change in growth rates. This feels a bit strange given that we do see relationships between things like length and aspect ratio and growth. I am currently thinking that this lack of relationship means that within an outcrossed line, size changes (for example) are associated with changes in growth rates. But, across outcrossed lines, some lines change their size a little whereas some lines change their sizes a lot and yet those two lines may show the same degree of growth rate change.

Conclusions

To wrap all of this up, what have we done and found?

In response to changes in temperature, the paramecia do exhibit phenotypic plasticity. In general, paramecium get shorter, less wide, skinnier and faster as temperatures increase, but this relationship is often nonlinear with evidence for a switch in this pattern at the highest temperatures. Moreover, changes in length and width and strongly correlated with one another as are changes in aspect ratio and speed. The correlations in traits between these groups is less strong which is also true for genetic variation among outcrossed lines. We also find that these morphological and movement changes are associated with changes in paramecium growth within outcrossed lines. That is, as the paramecia within outcrossed lines get smaller, skinnier, and faster, they also tend to have higher growth rates. However, across outcrossed lines, we do not find a relationship between the extent to which morphology and movement change and the total extent to which paramecium growth rates change either from a univariate or multivariate view of their morphological and movement changes. Looking forward to folks' thoughts about this.