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 partly be attributable 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 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.

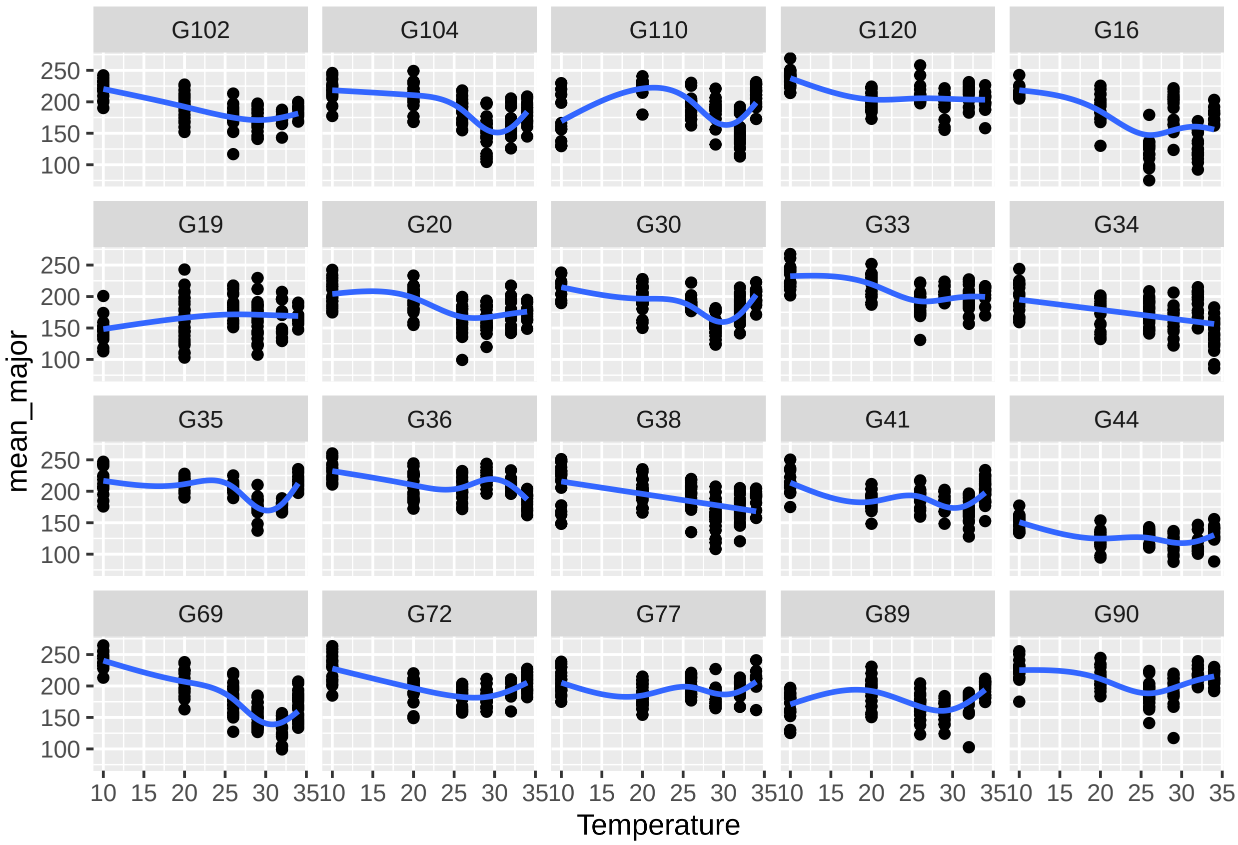
```
morph_data <- read.csv('Plast_StartPop_Data.csv')

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 flexibly fitting relationships when they are expected to be nonlinear and potentially so in a way that is not necessarily known a priori. Furthermore, it is easy to use predictions from these models to characterize the plasticity of different outcrossed lines. Below, I will run through each of the phenotypes.

Length

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

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



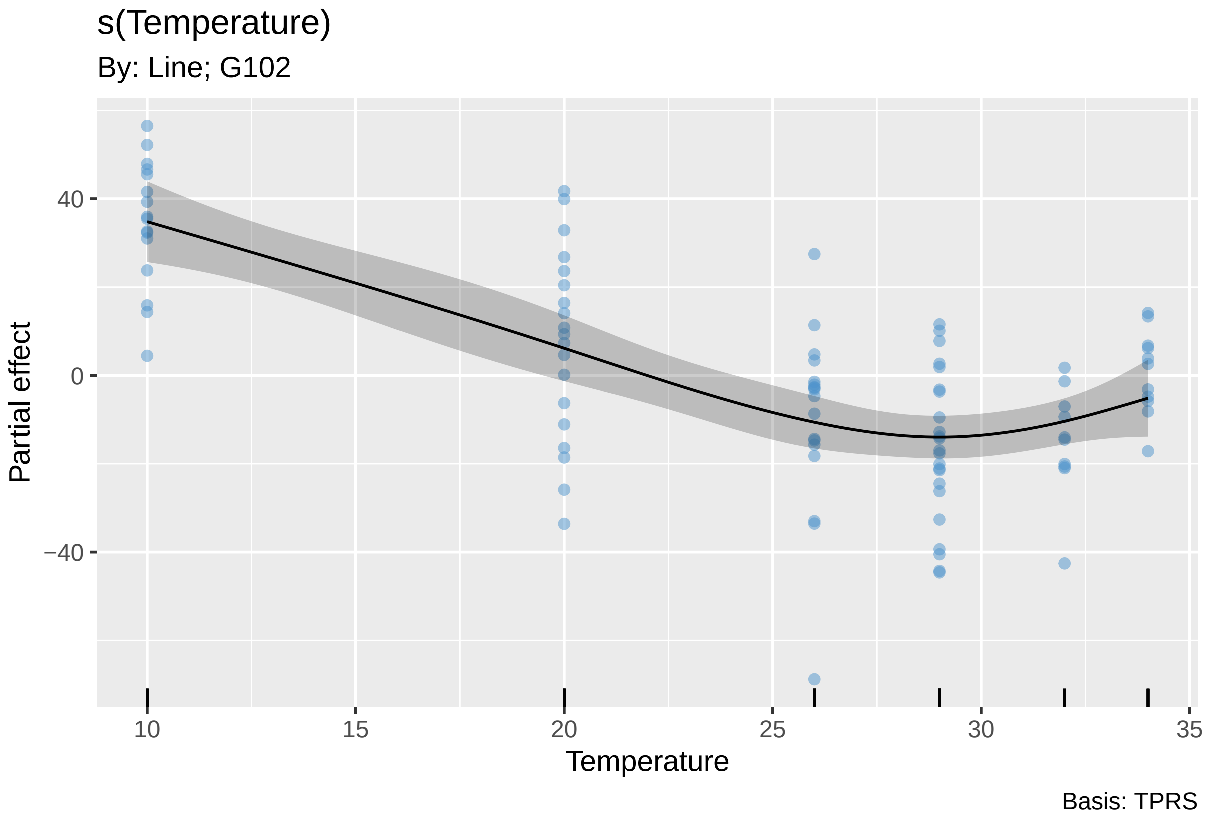
There is a lot going on here, but, what we see generally is a tendency towards smaller lengths (mean major axis) as temperature increases with a slight increase in length at the highest temperature. Now we can run the formal GAMs and look at the resulting summary and plots.

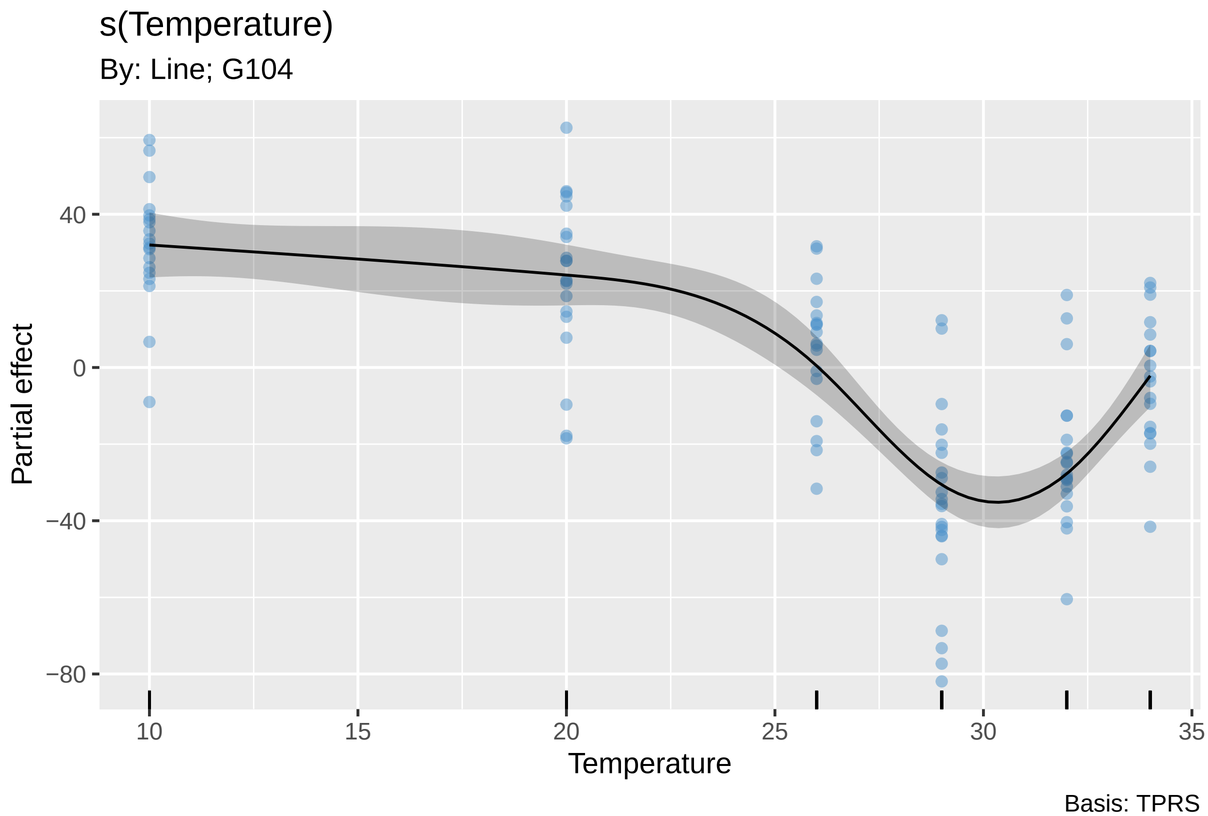
```r
gam_length <- gam(formula = mean_major ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = mor

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 .
```

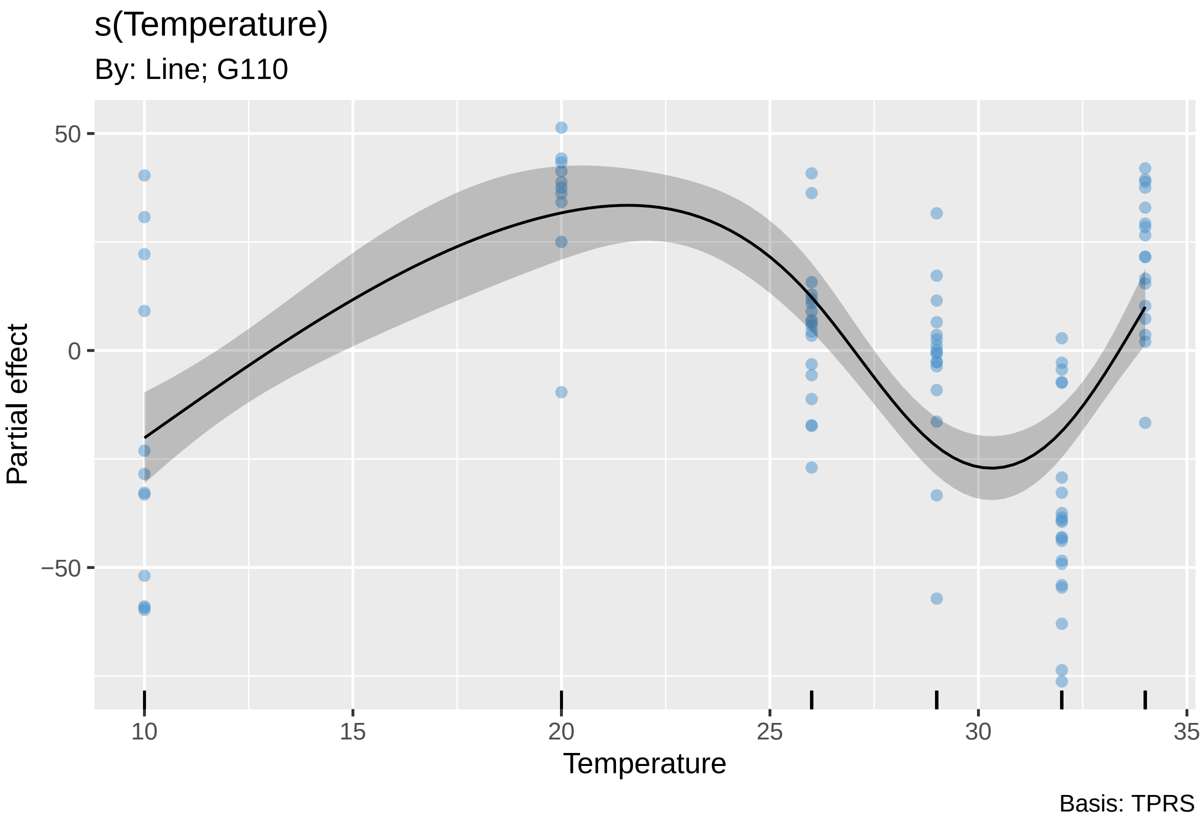
```
## LineG30        2.630      2.916   0.902 0.367243
## LineG33       20.876      2.947   7.084 1.93e-12 ***
## LineG34      -13.682      2.865  -4.775 1.93e-06 ***
## LineG35       14.506      3.140   4.619 4.09e-06 ***
## LineG36       24.990      3.025   8.262 2.57e-16 ***
## LineG38        2.130      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
```

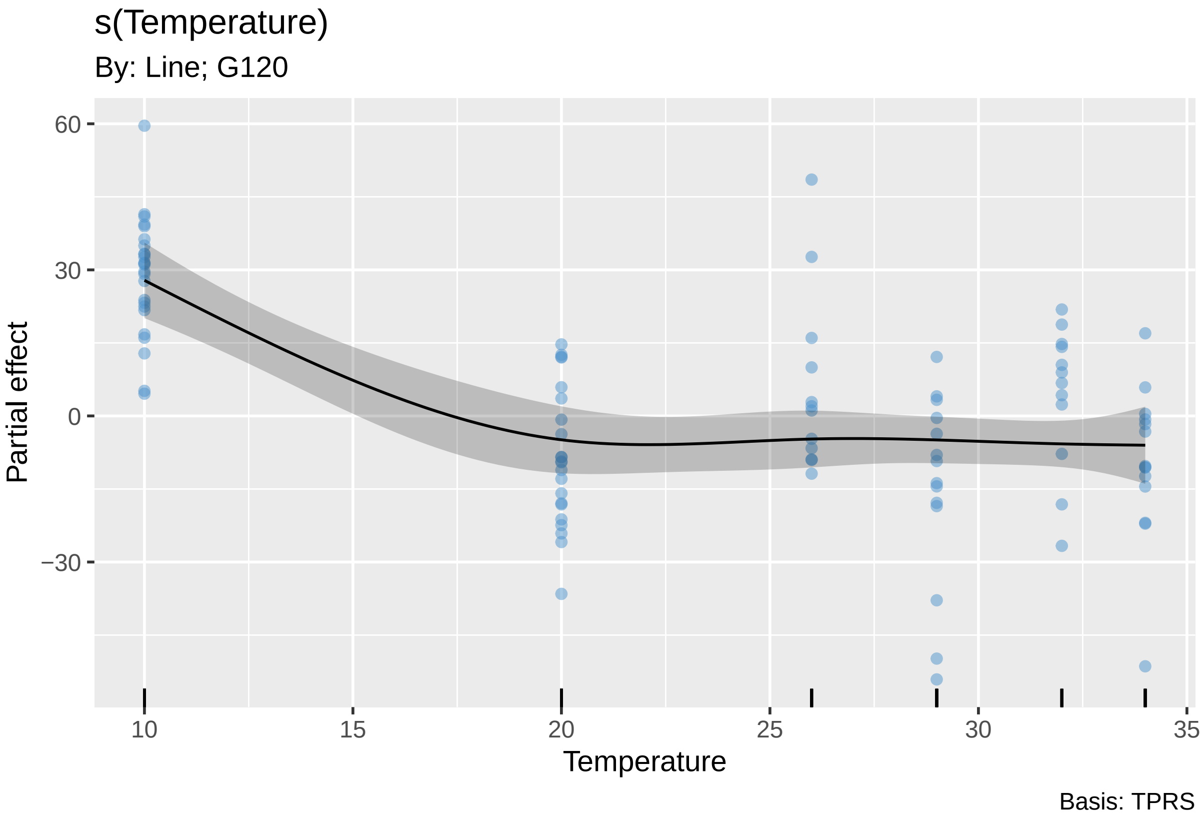
```r
for (i in 1:length(smooths(gam_length))) {
  out <- draw(gam_length, select = i, residuals = TRUE)
  print(out) }
```


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

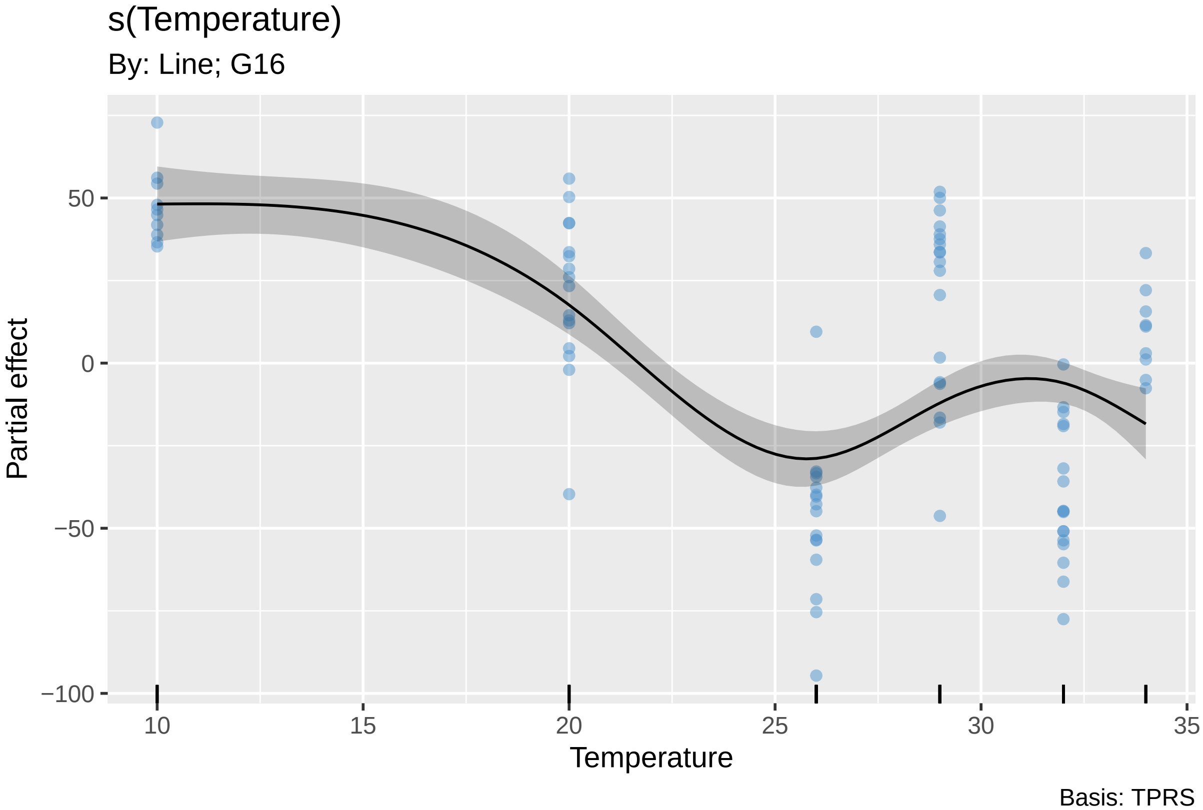


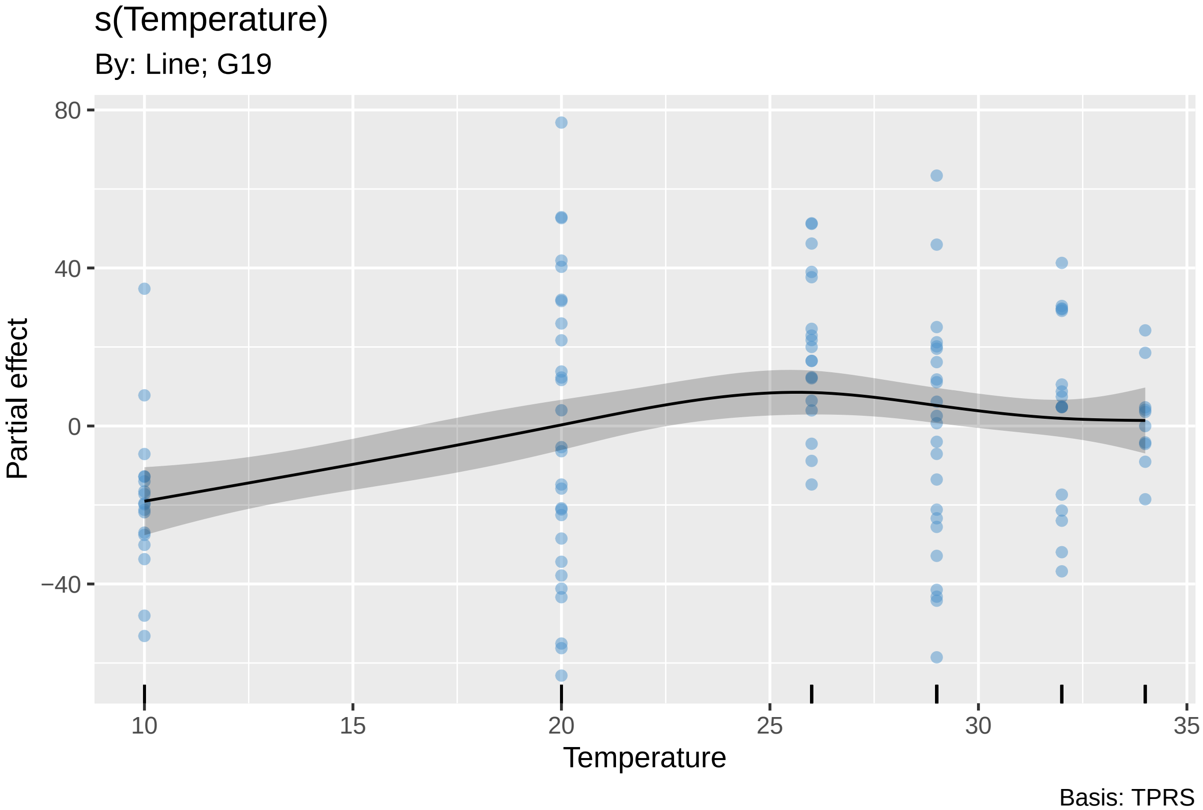
s(Temperature)
By: Line; G104
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS


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

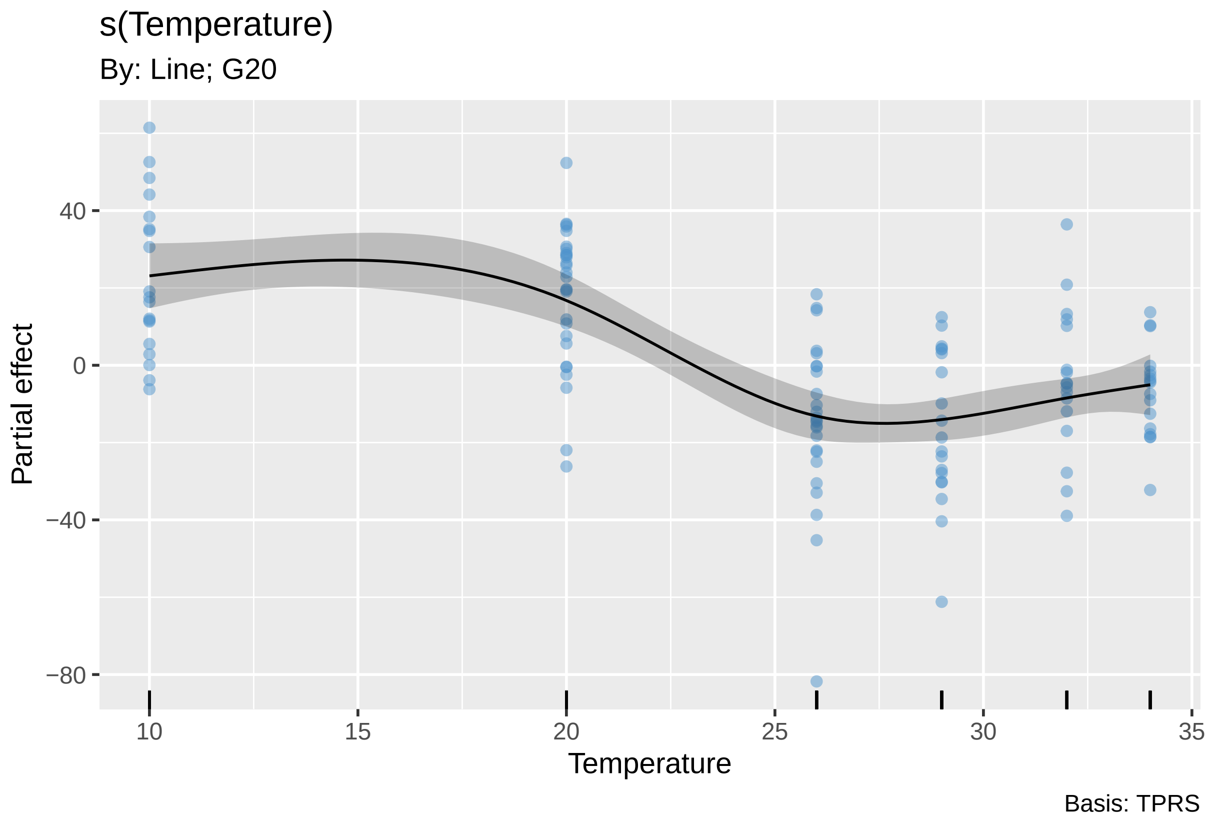


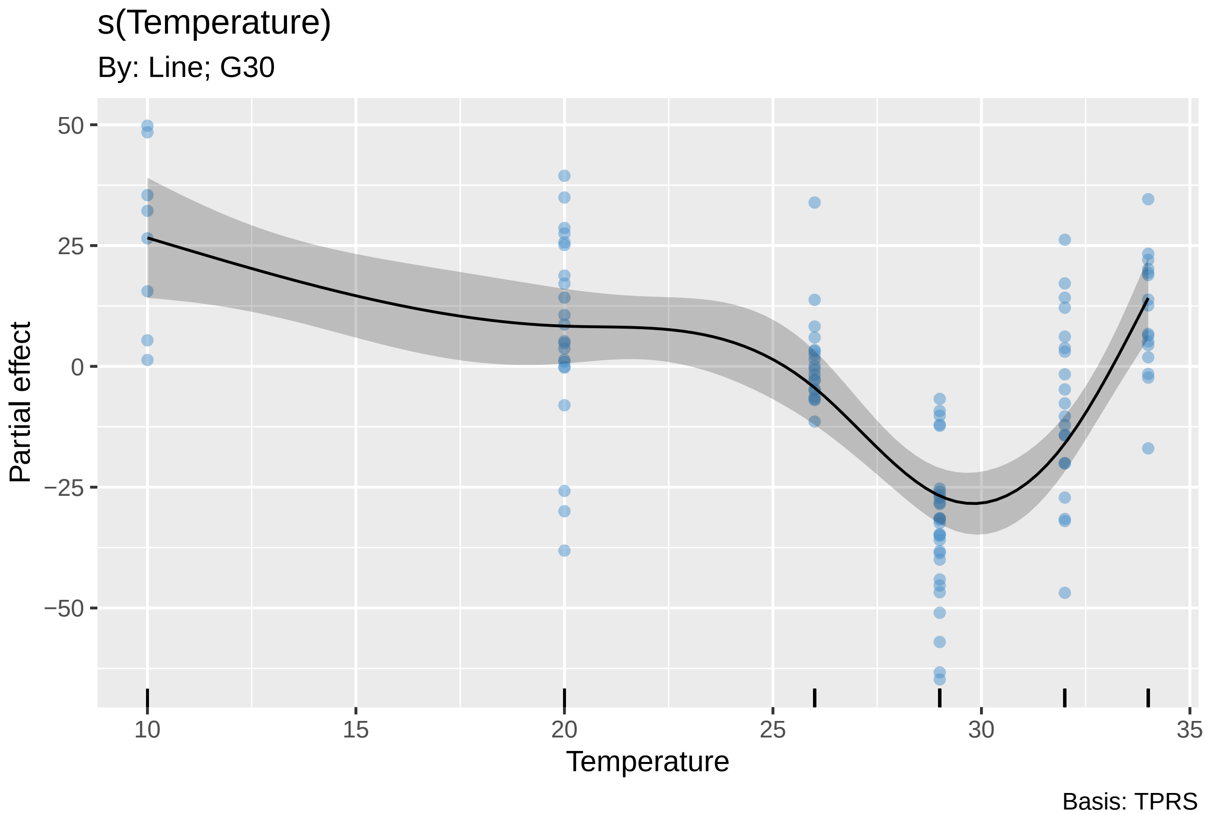
s(Temperature)
By: Line; G120
Partial effect
60
30
0
−30
10
15
20
25
30
35
Temperature
Basis: TPRS

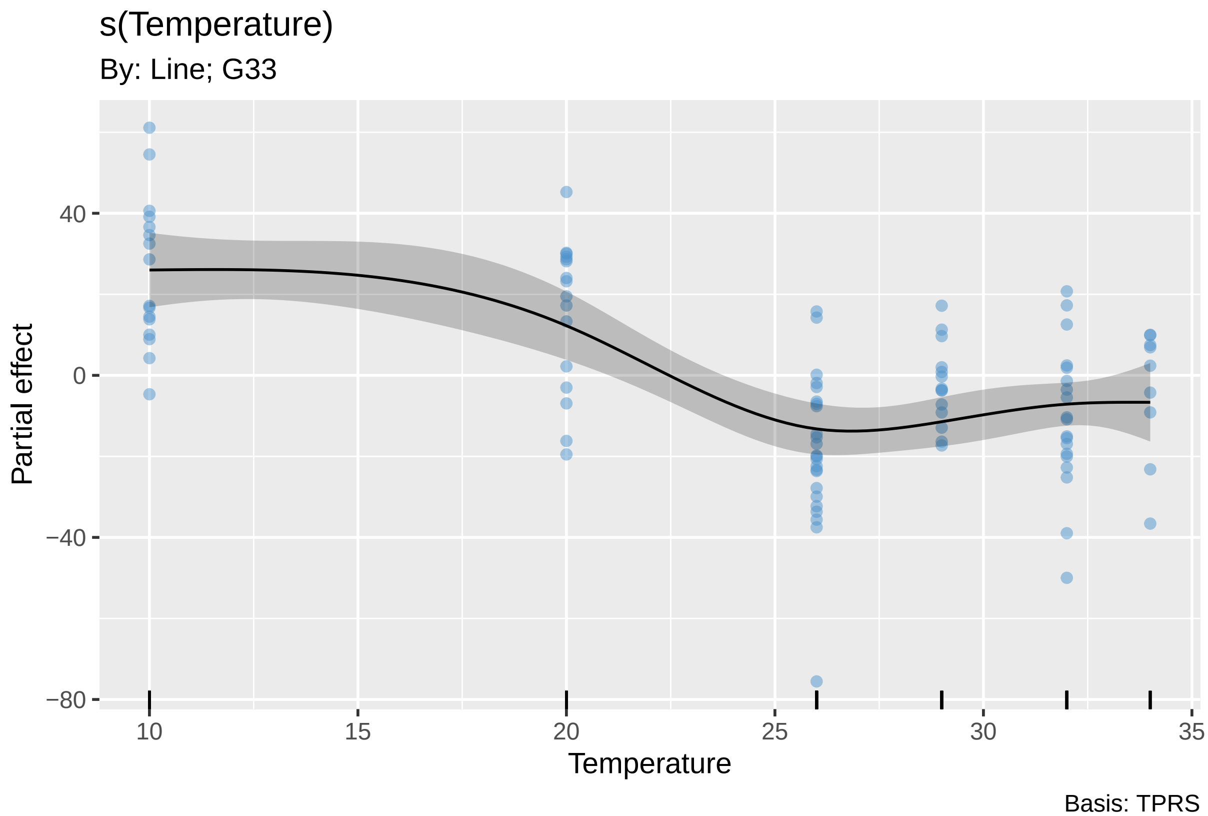

s(Temperature)
By: Line; G16
Partial effect
50
0
−50
−100
10
15
20
25
30
35
Temperature
Basis: TPRS

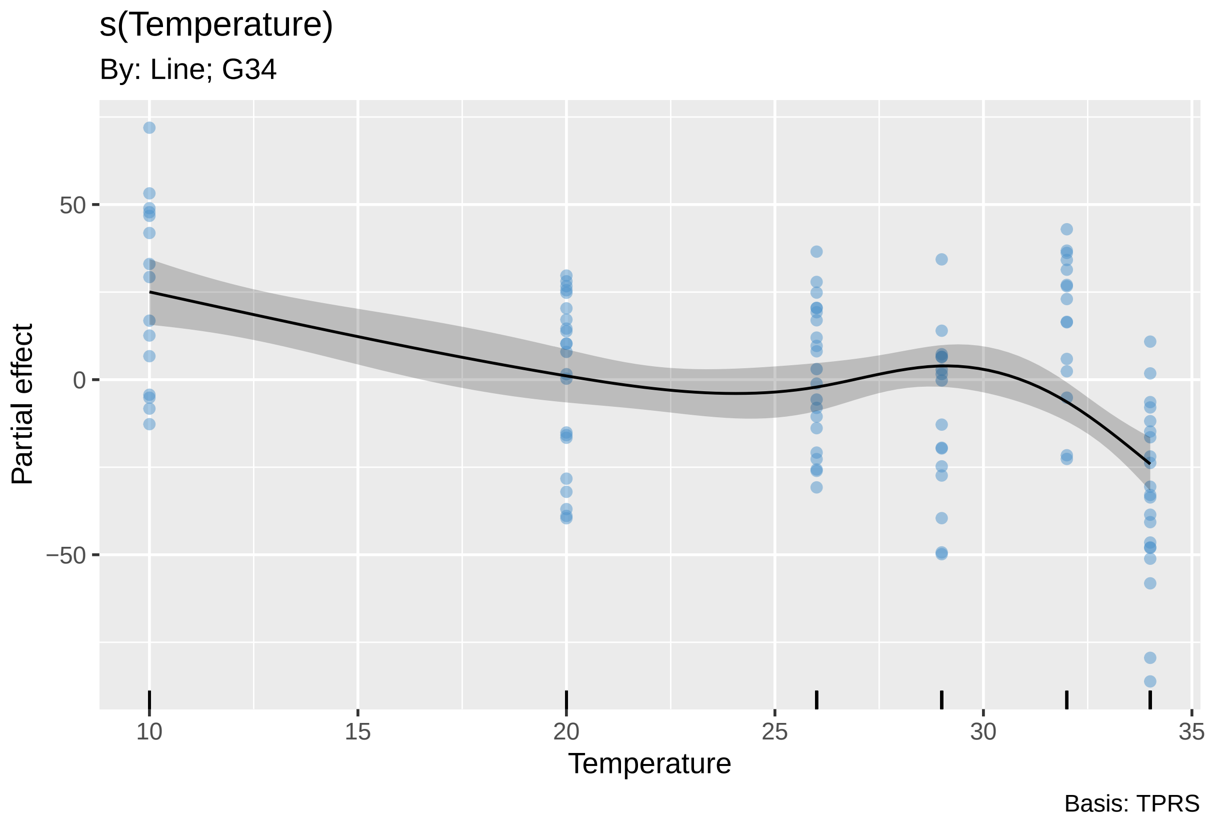


s(Temperature)
By: Line; G19
Partial effect
80
40
0
−40
10
15
20
25
30
35
Temperature
Basis: TPRS

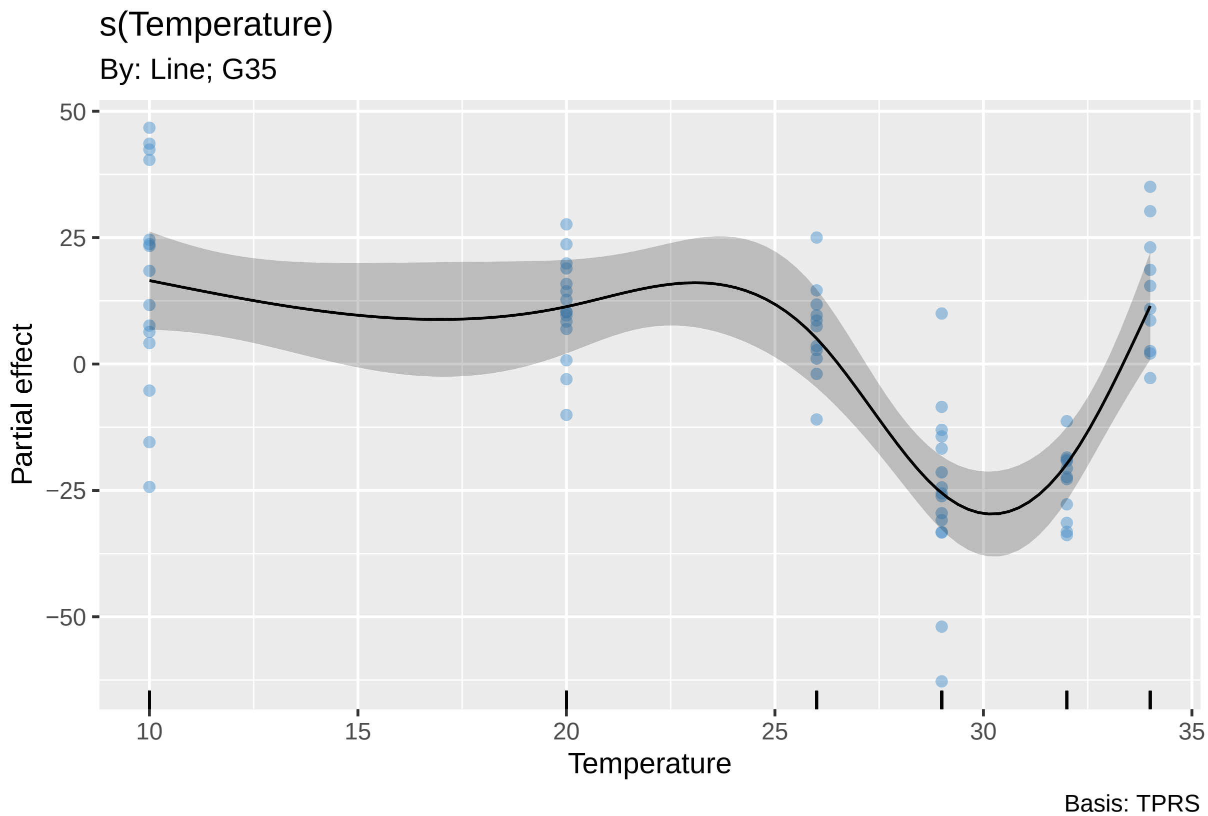

s(Temperature)
By: Line; G20
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS


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

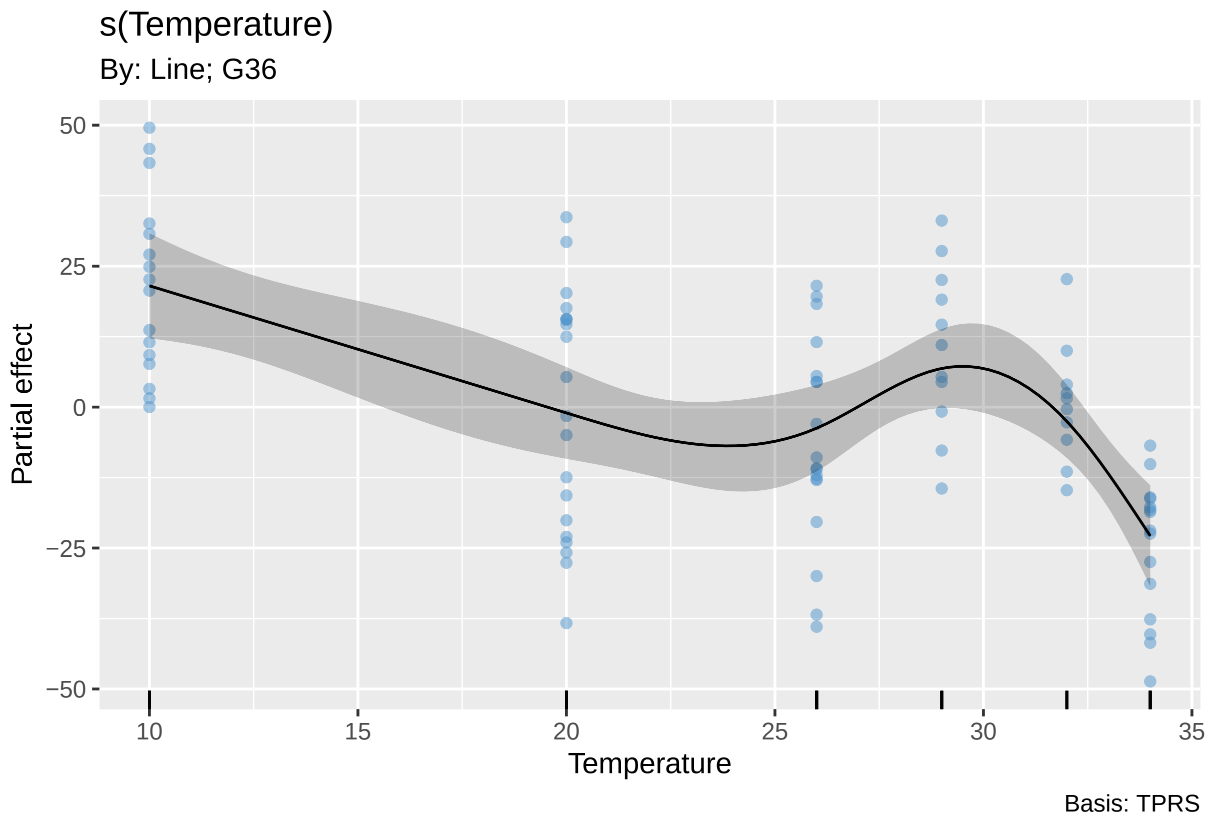

s(Temperature)
By: Line; G33
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS

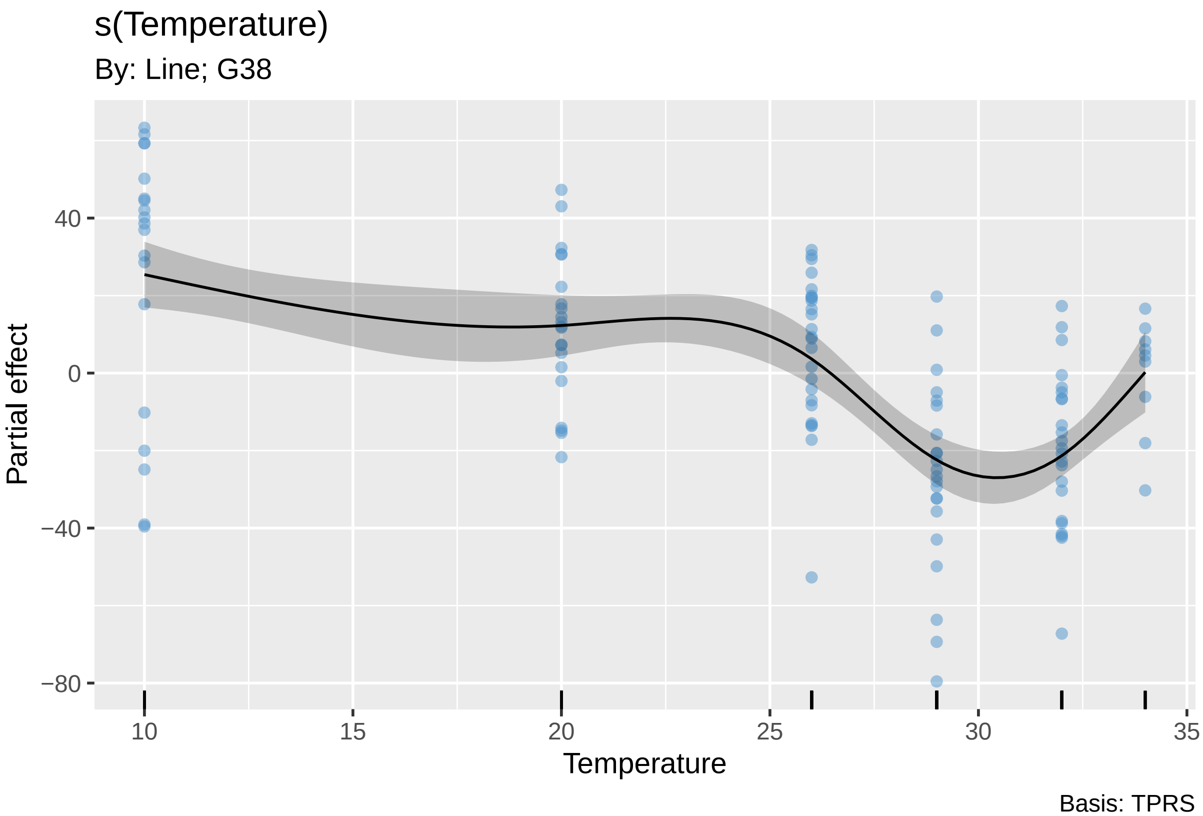


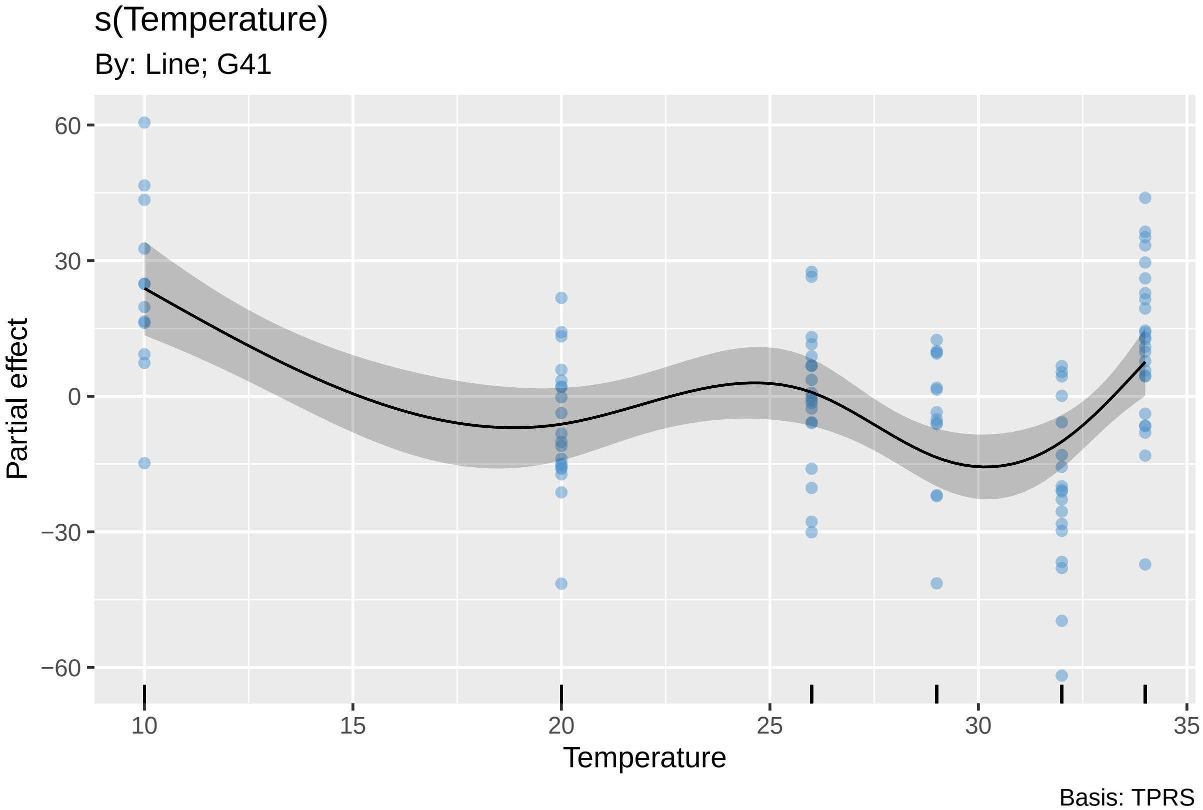
s(Temperature)
By: Line; G34
Partial effect
50
0
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

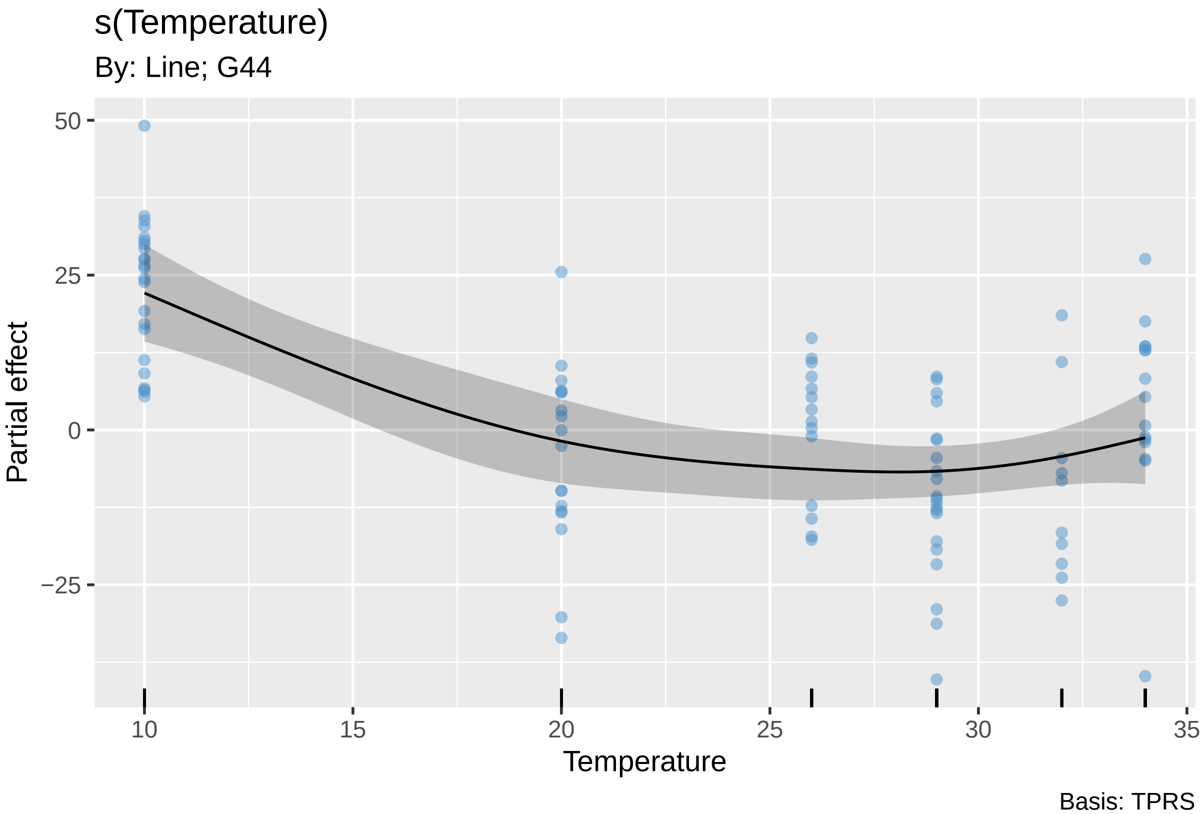


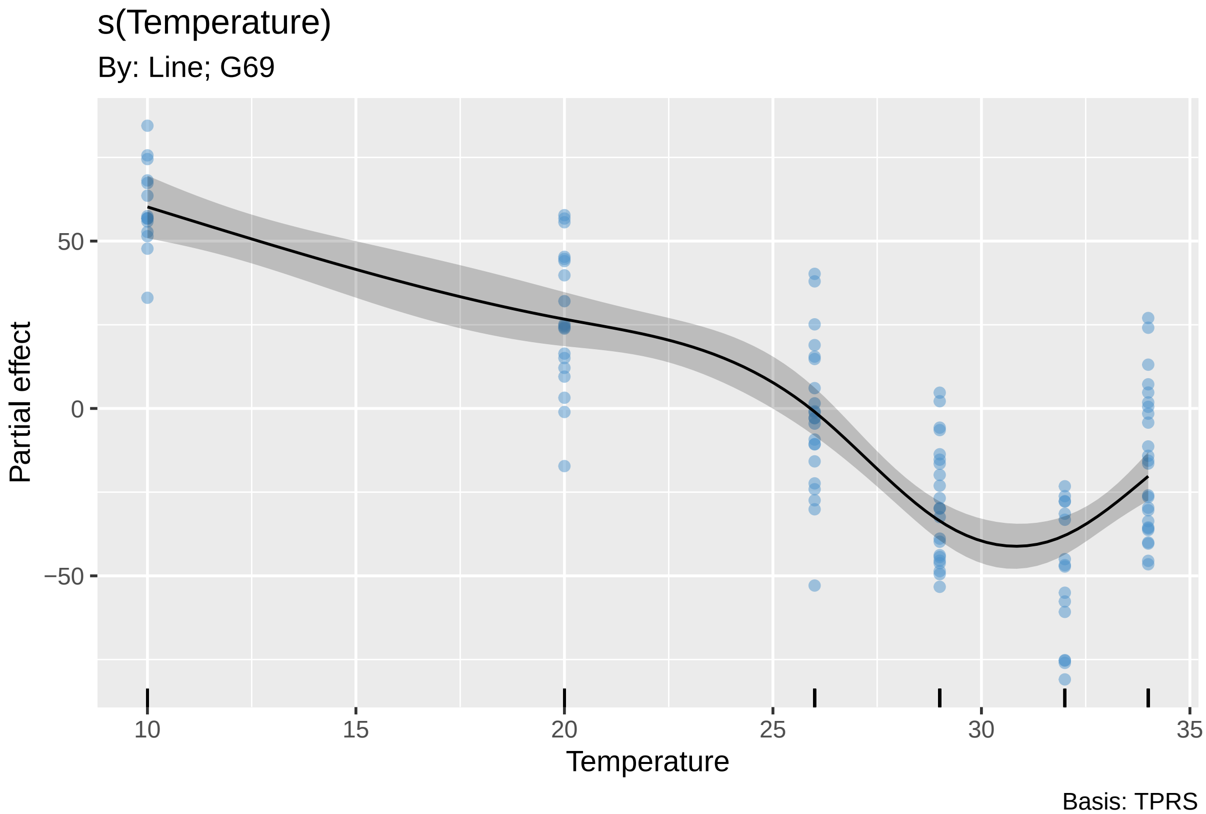
s(Temperature)
By: Line; G35
Partial effect
50
25
0
−25
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

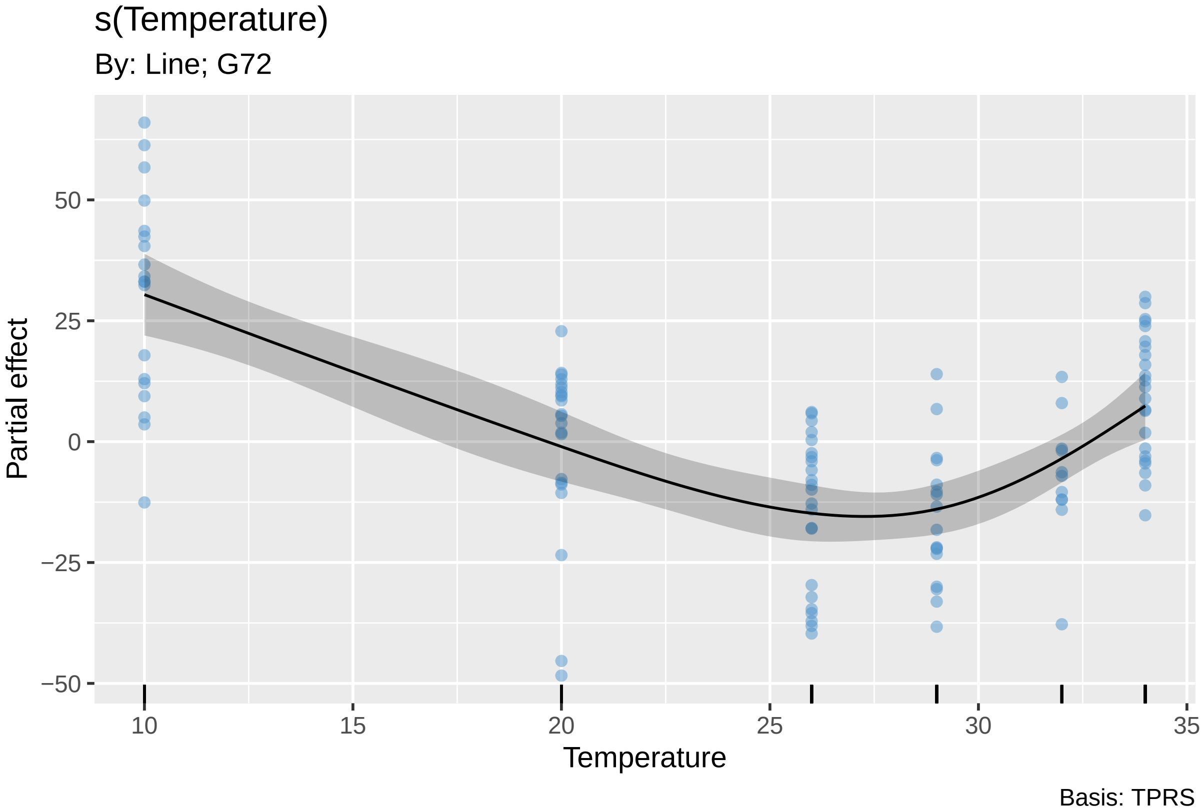

s(Temperature)
By: Line; G36
Partial effect
50
25
0
−25
−50
10
15
20
25
30
35
Temperature
Basis: TPRS

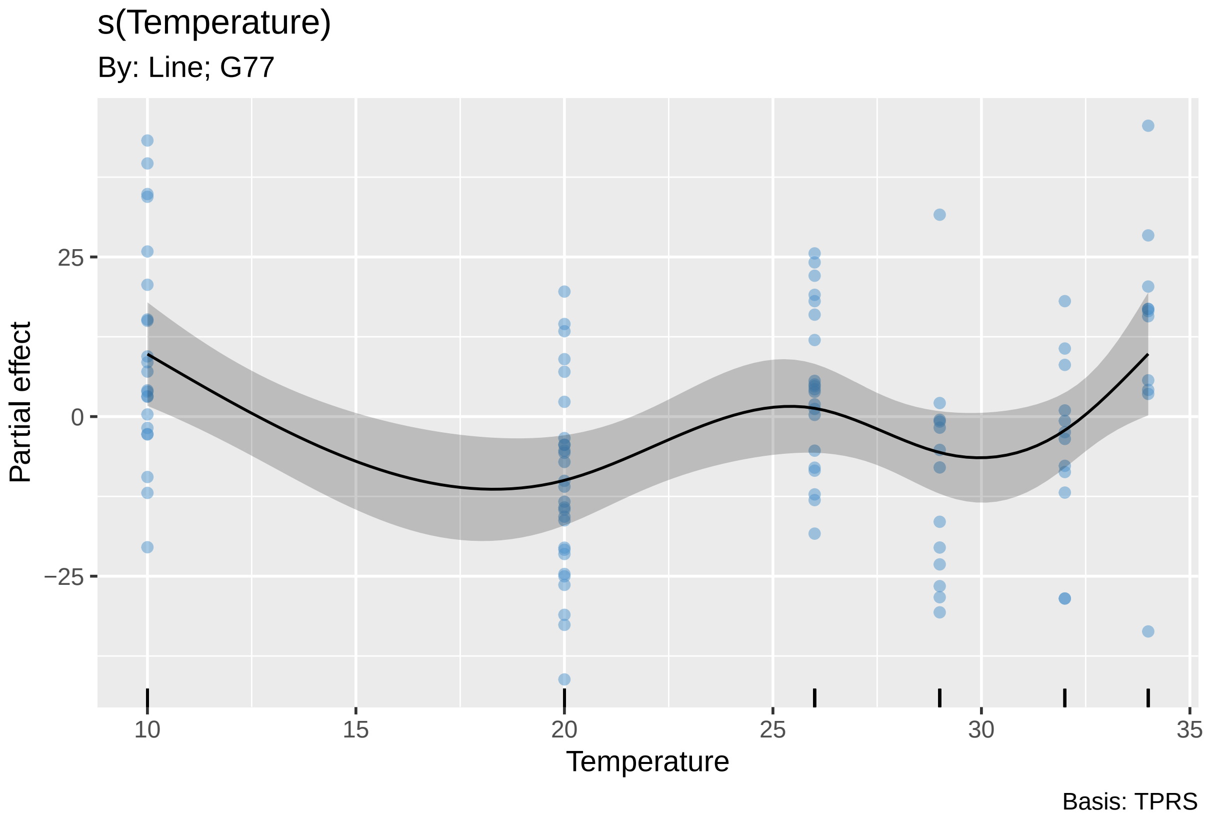

s(Temperature)
By: Line; G38
Partial effect
40
0
−40
−80
10
15
20
25
30
35
Temperature
Basis: TPRS


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

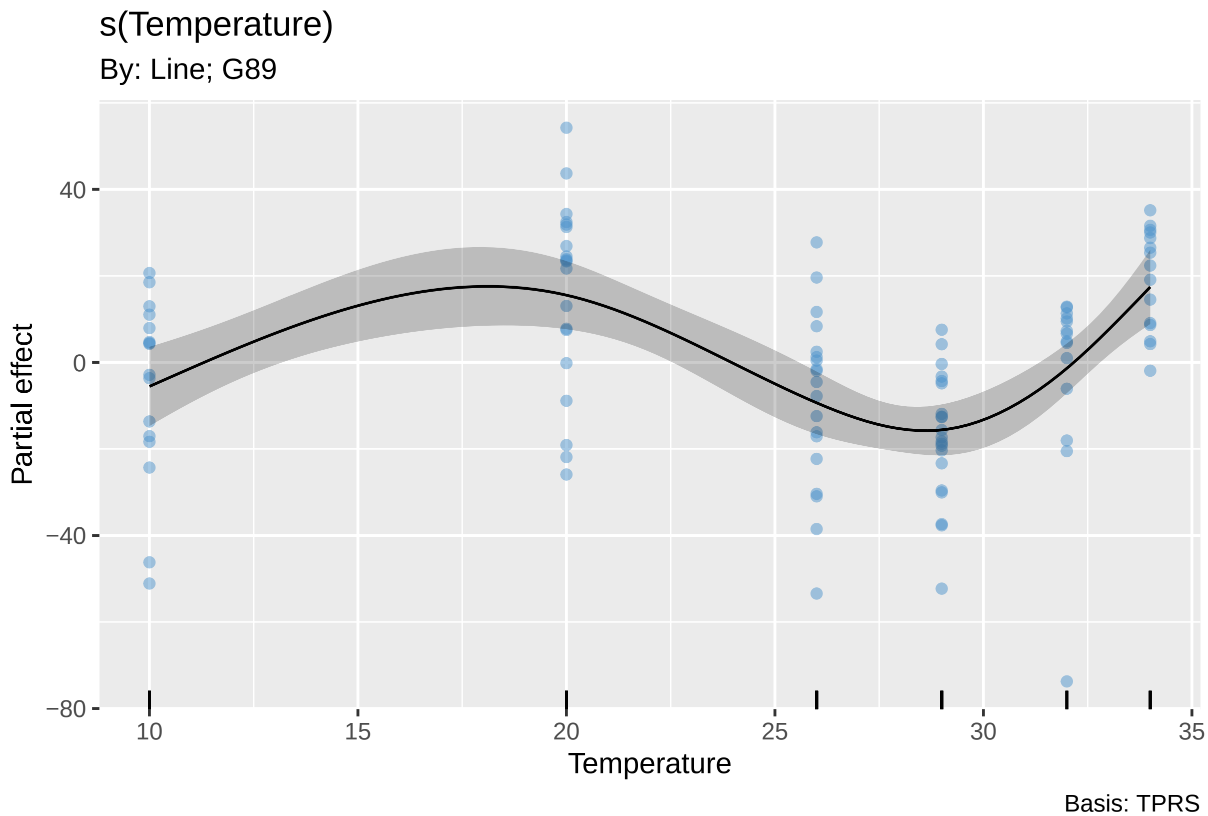

s(Temperature)
By: Line; G44
Partial effect
50
25
0
−25
10
15
20
25
30
35
Temperature
Basis: TPRS

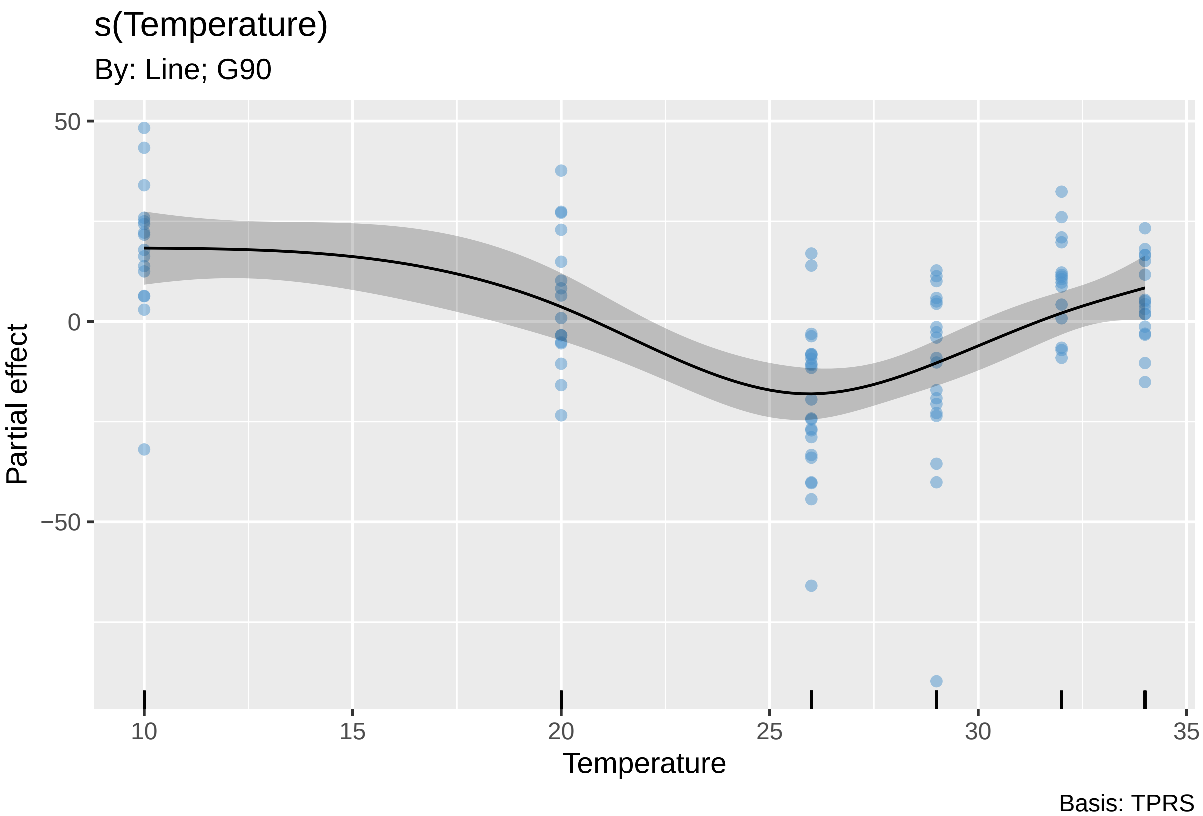

s(Temperature)
By: Line; G69
Partial effect
50
0
−50
10
15
20
25
30
35
Temperature
Basis: TPRS


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)
```

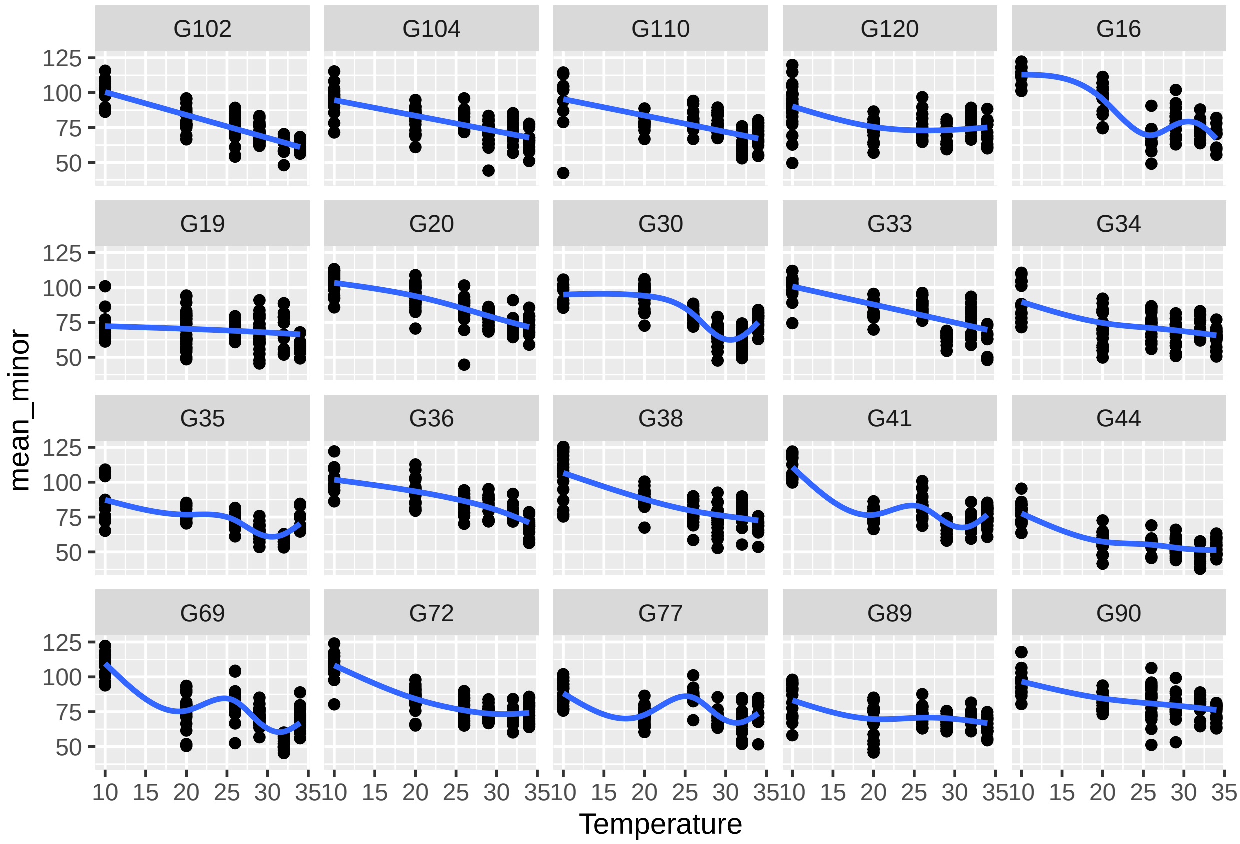
```
##                        df      AIC
## gam_length       95.56545 18719.53
## gam_length_noint 24.99633 19293.96
```

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)
```



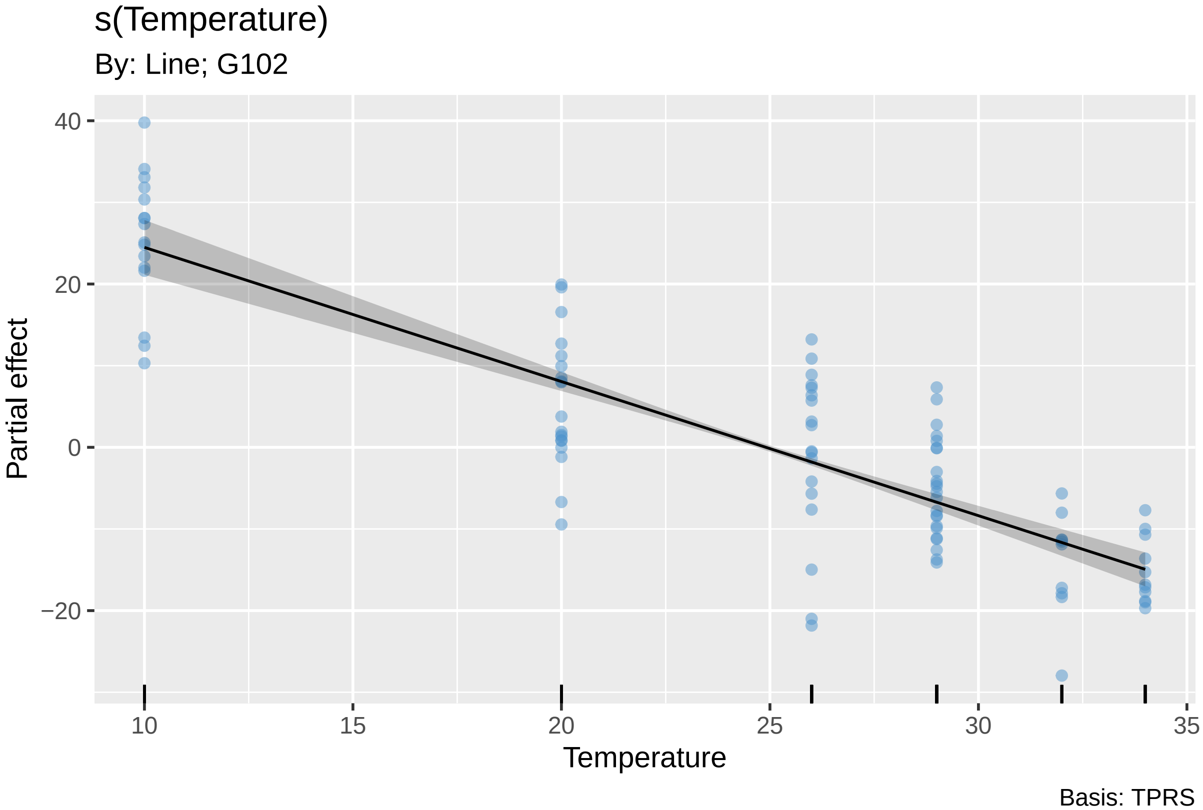
In general, we see a similar trend as the one we saw with length – paramecia are generally getting less wide as temperatures increase. Again, we will run GAMs and look at the results.

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

summary(gam_width)
```

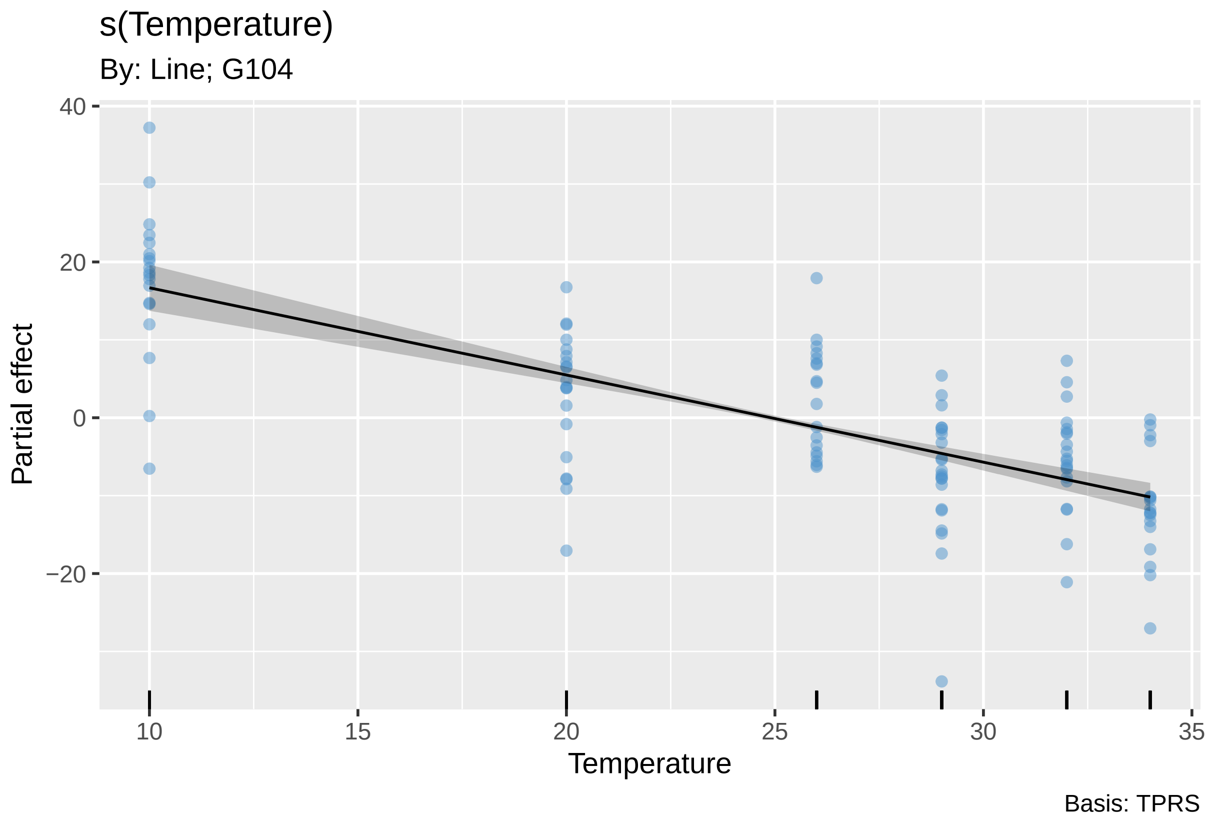
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean_minor ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  76.0081     0.8863  85.761  < 2e-16 ***
## LineG104      2.0314     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 **
```

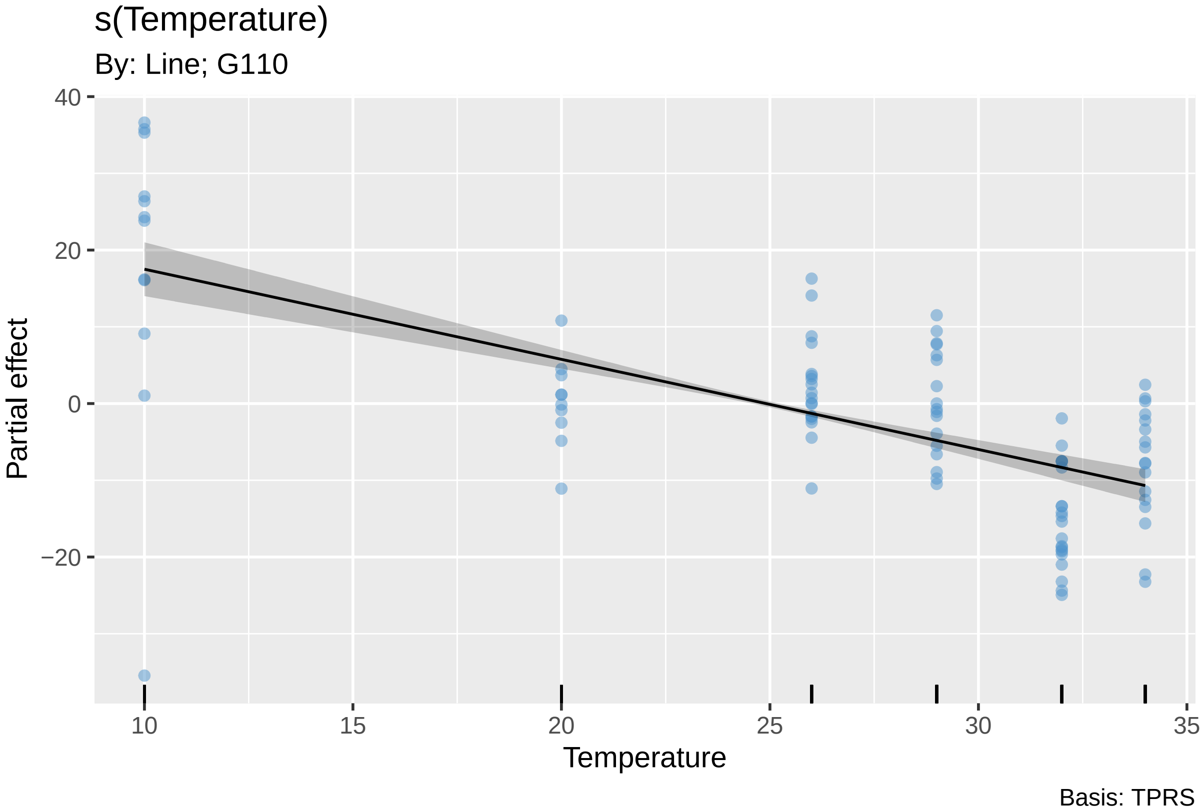
```
## LineG36       9.5426     1.2808   7.451 1.37e-13 ***
## LineG38       7.0089     1.1971   5.855 5.55e-09 ***
## LineG41       4.8101     1.2424   3.872 0.000111 ***
## LineG44     -18.3084     1.2482 -14.668  < 2e-16 ***
## LineG69       1.3793     1.2011   1.148 0.250955
## LineG72       5.6490     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
```

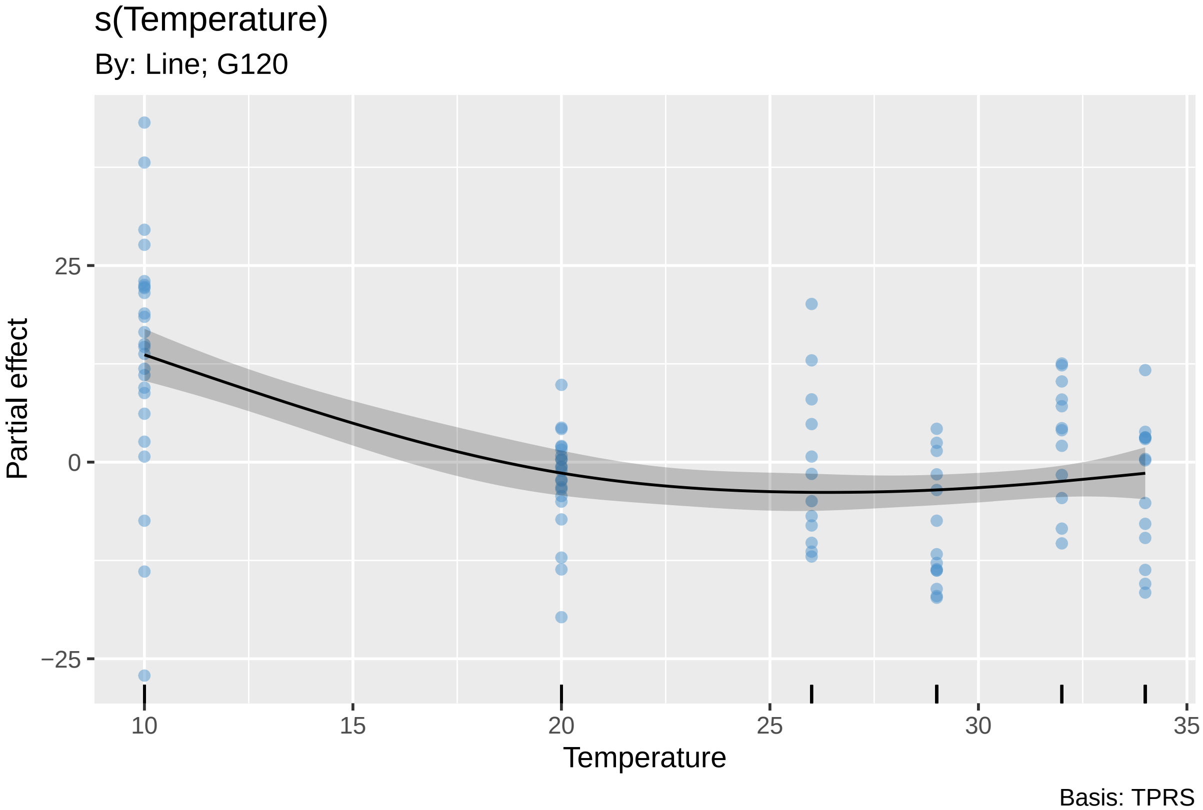
```
for (i in 1:length(smooths(gam_width))) {
  out <- draw(gam_width, select = i, residuals = TRUE)
  print(out) }
```

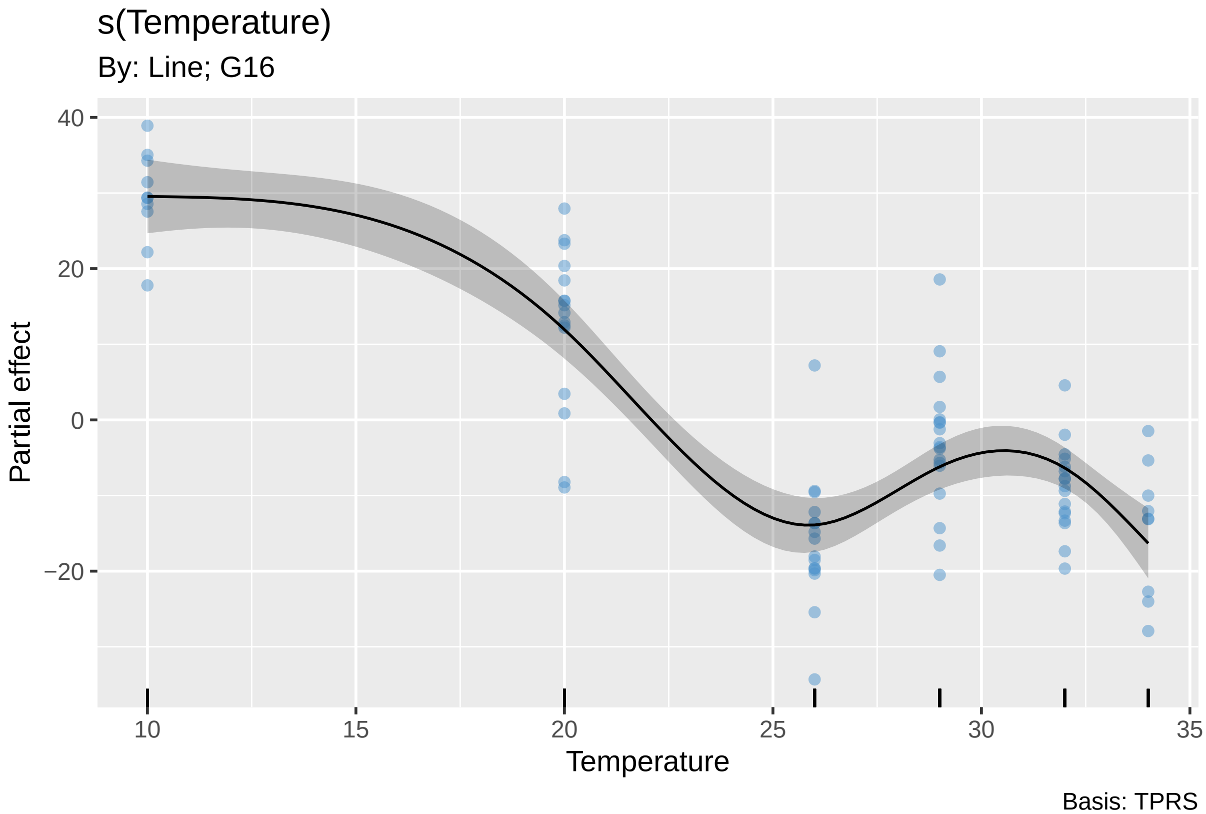


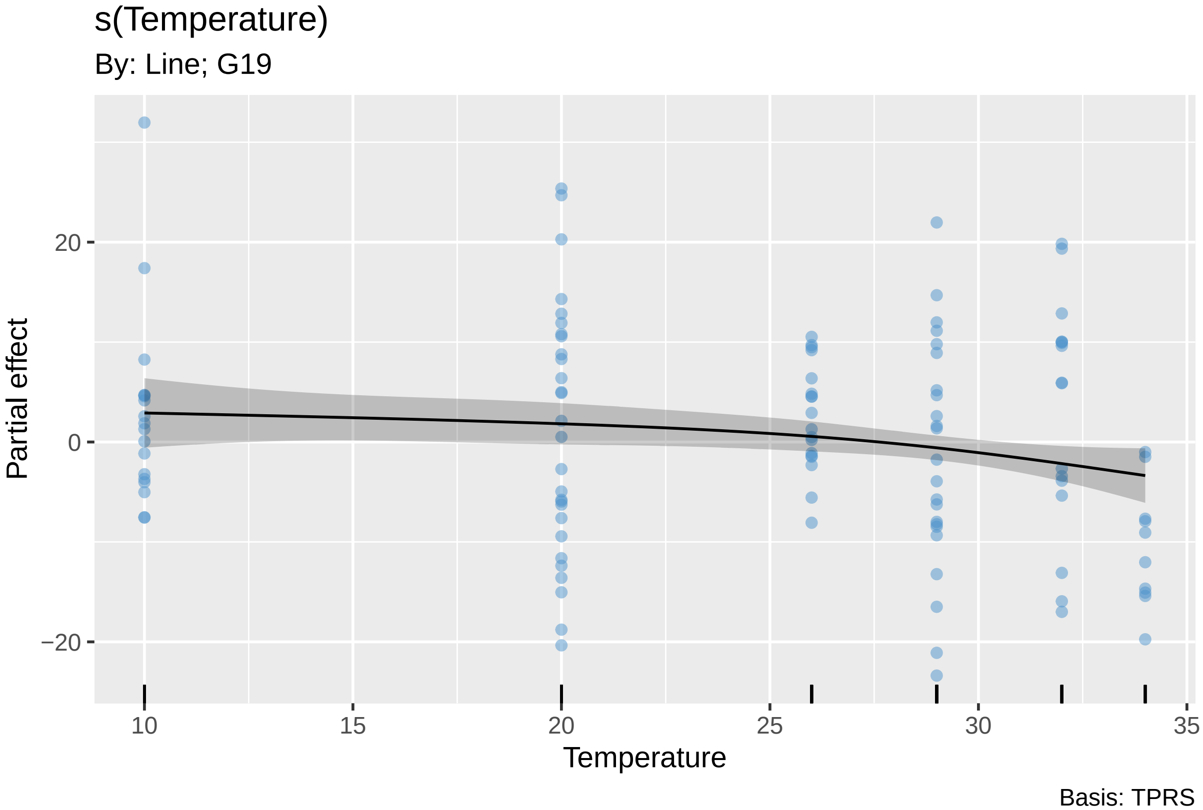
s(Temperature)
By: Line; G102
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

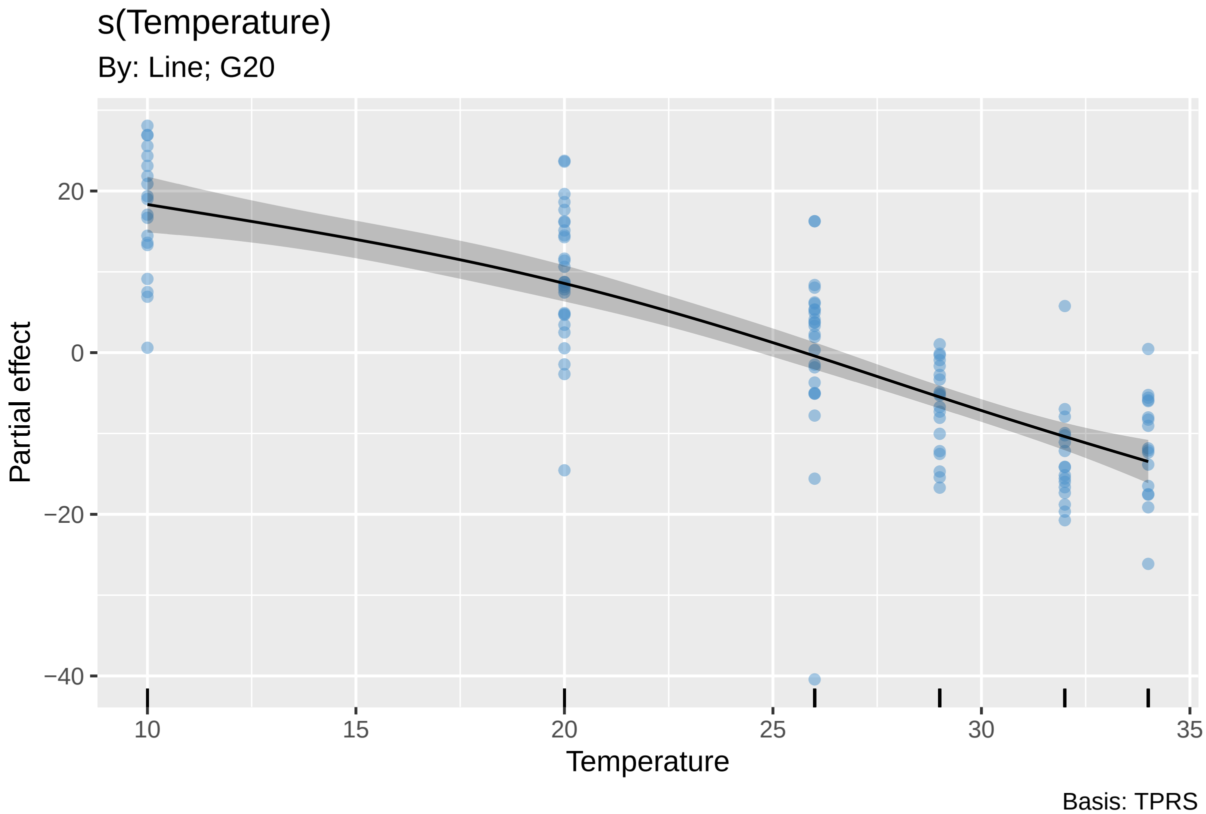

s(Temperature)
By: Line; G104
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

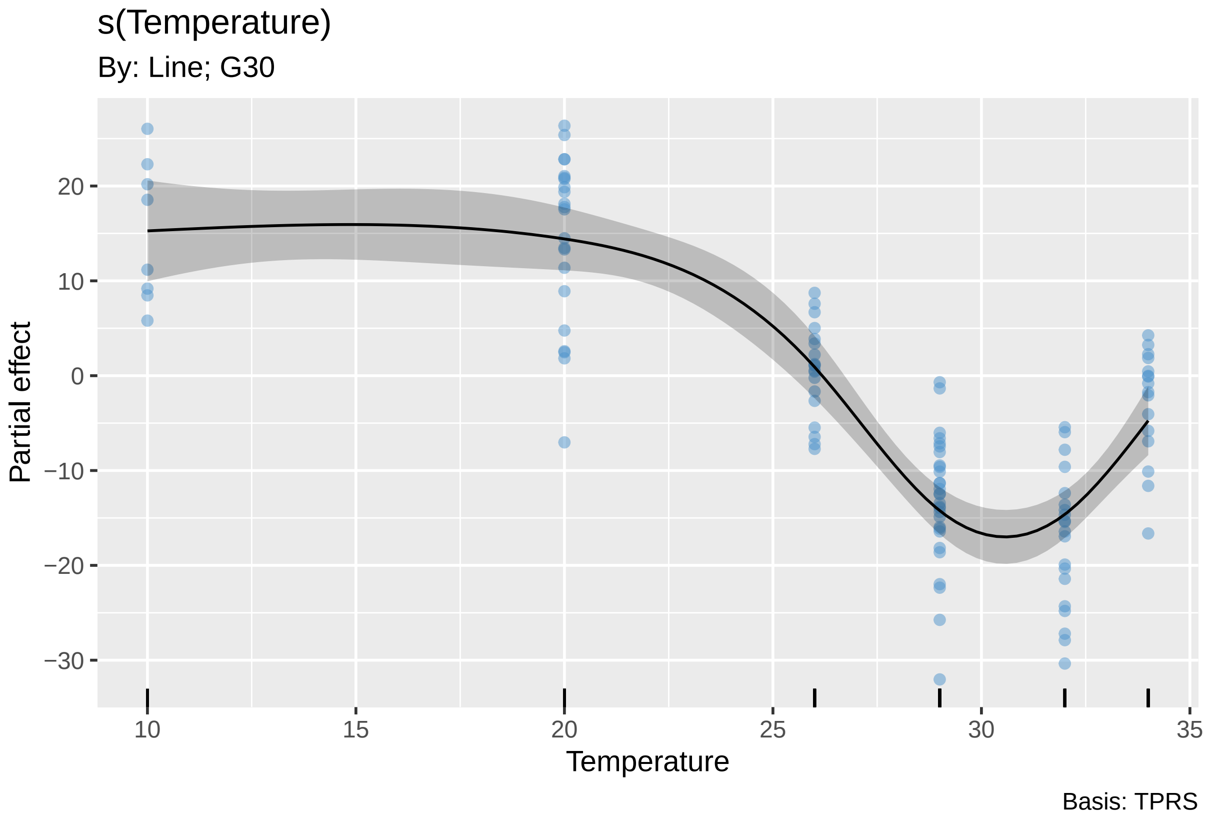

s(Temperature)
By: Line; G110
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


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

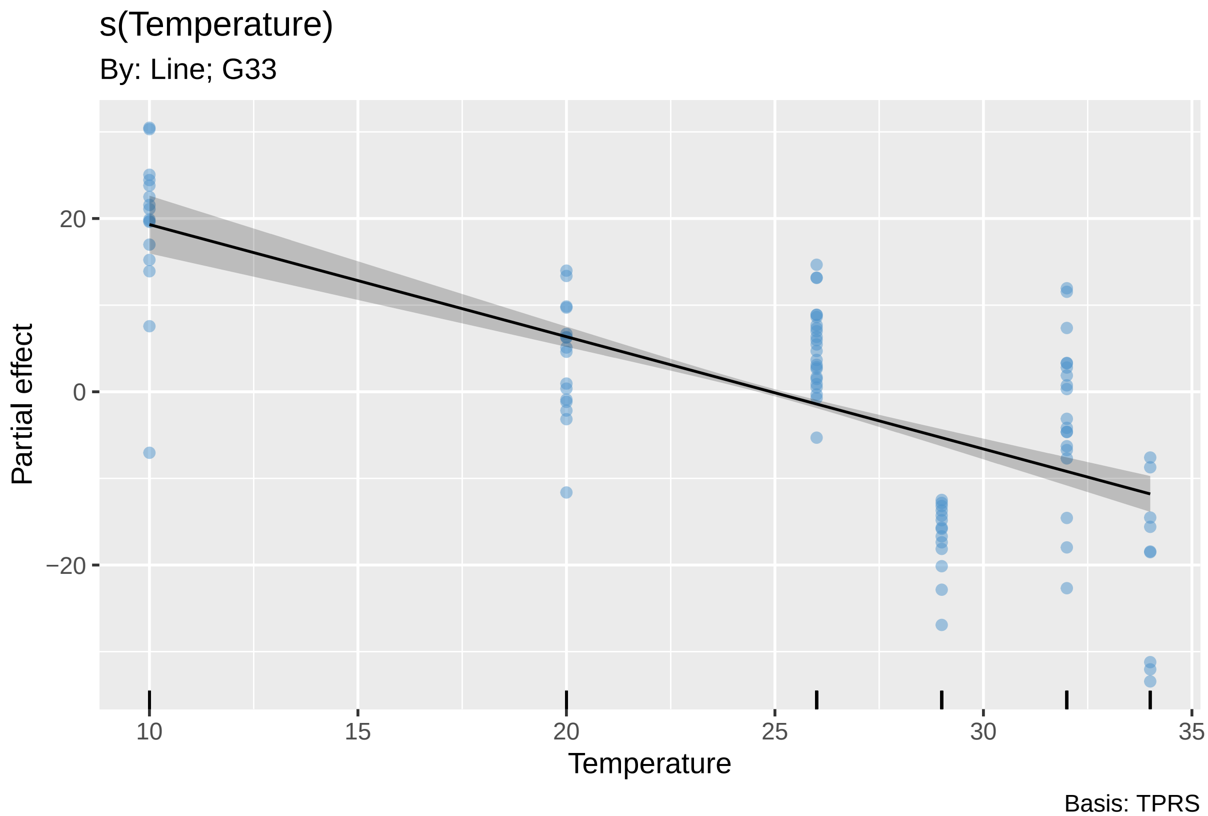

s(Temperature)
By: Line; G16
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

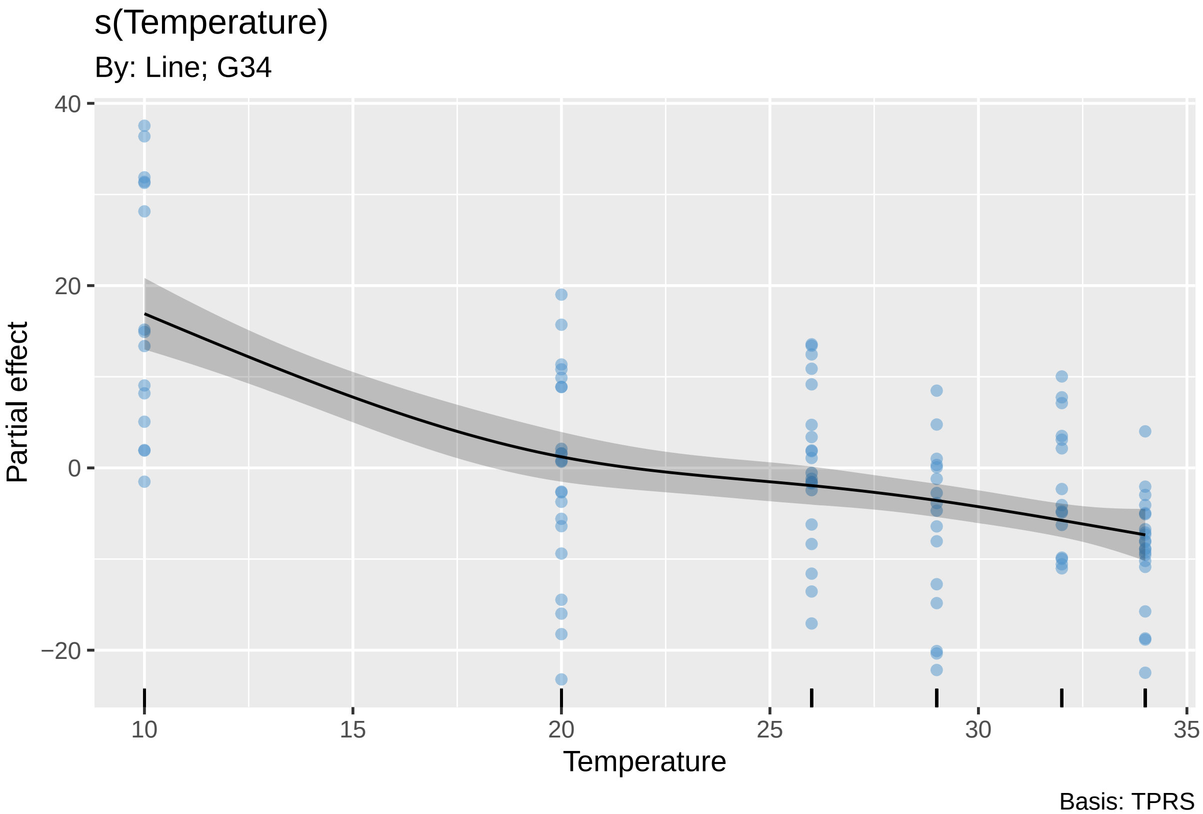

s(Temperature)
By: Line; G19
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

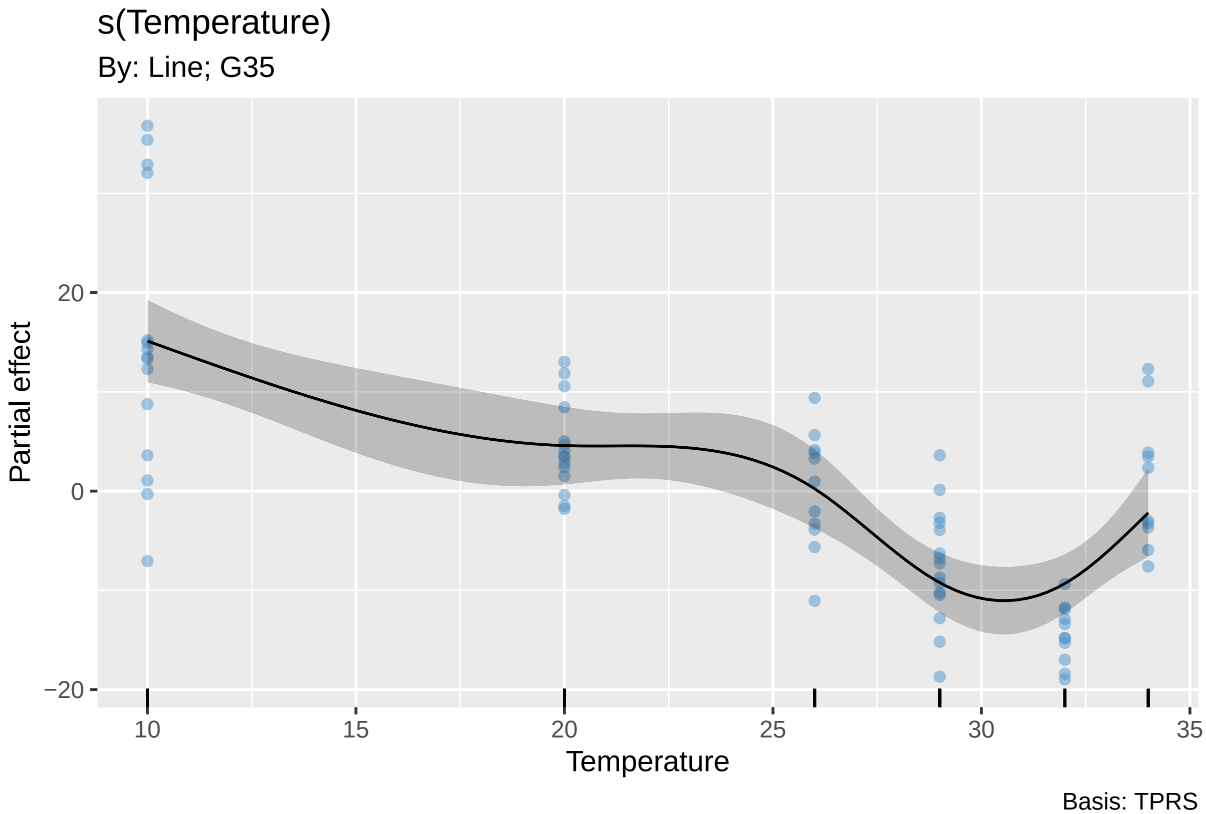

s(Temperature)
By: Line; G20
Partial effect
20
0
−20
−40
10
15
20
25
30
35
Temperature
Basis: TPRS

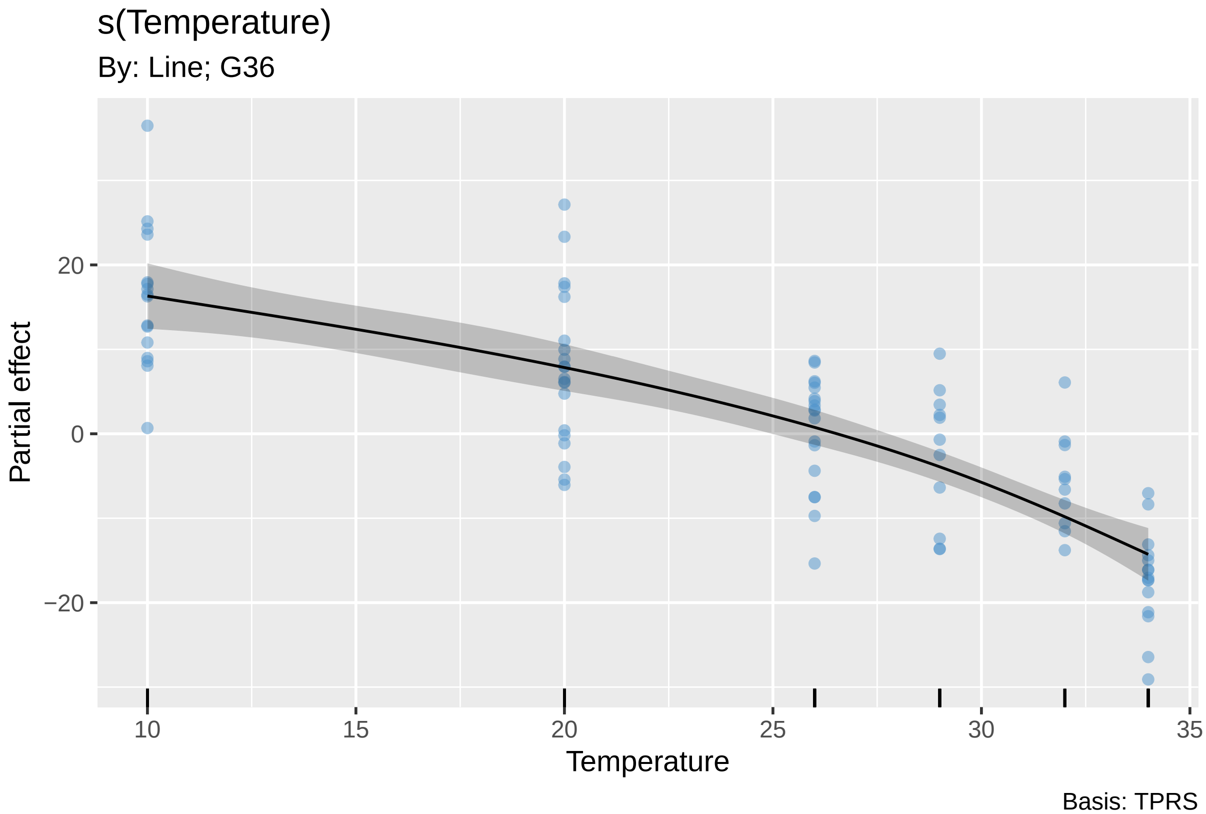


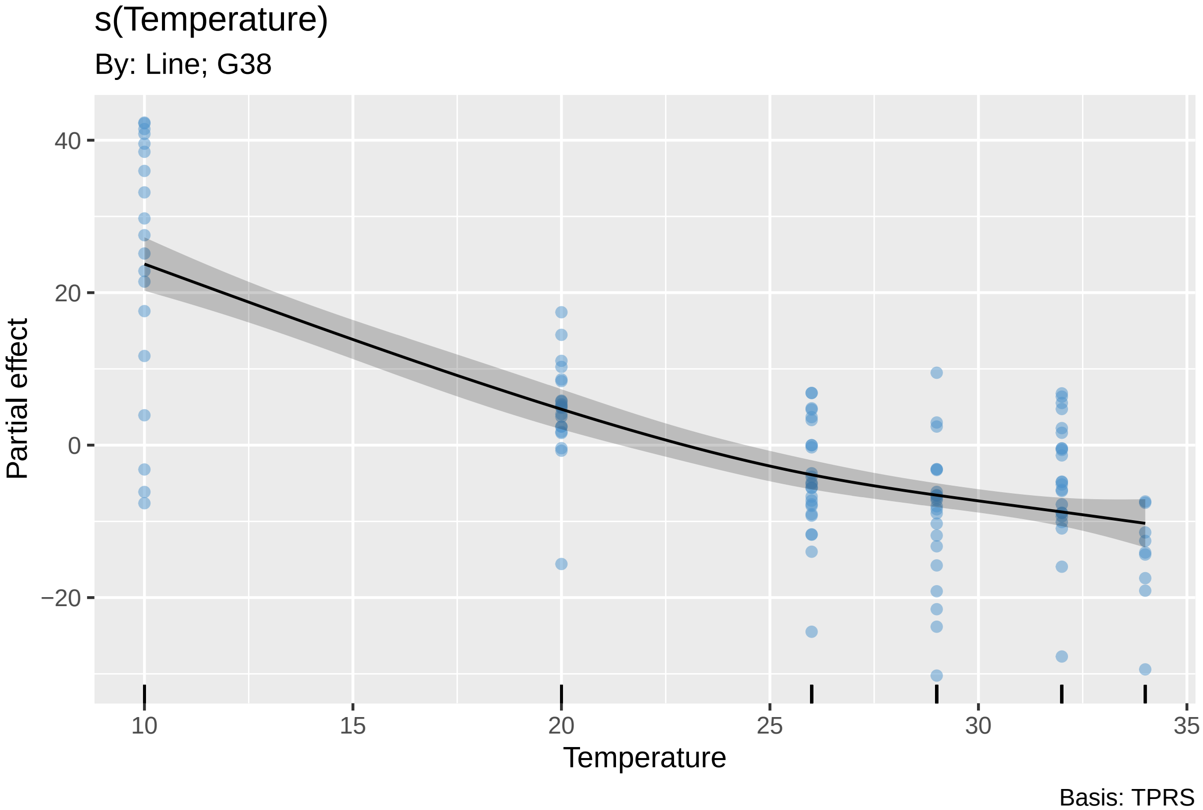
s(Temperature)
By: Line; G30
Partial effect
20
10
0
−10
−20
−30
10
15
20
25
30
35
Temperature
Basis: TPRS

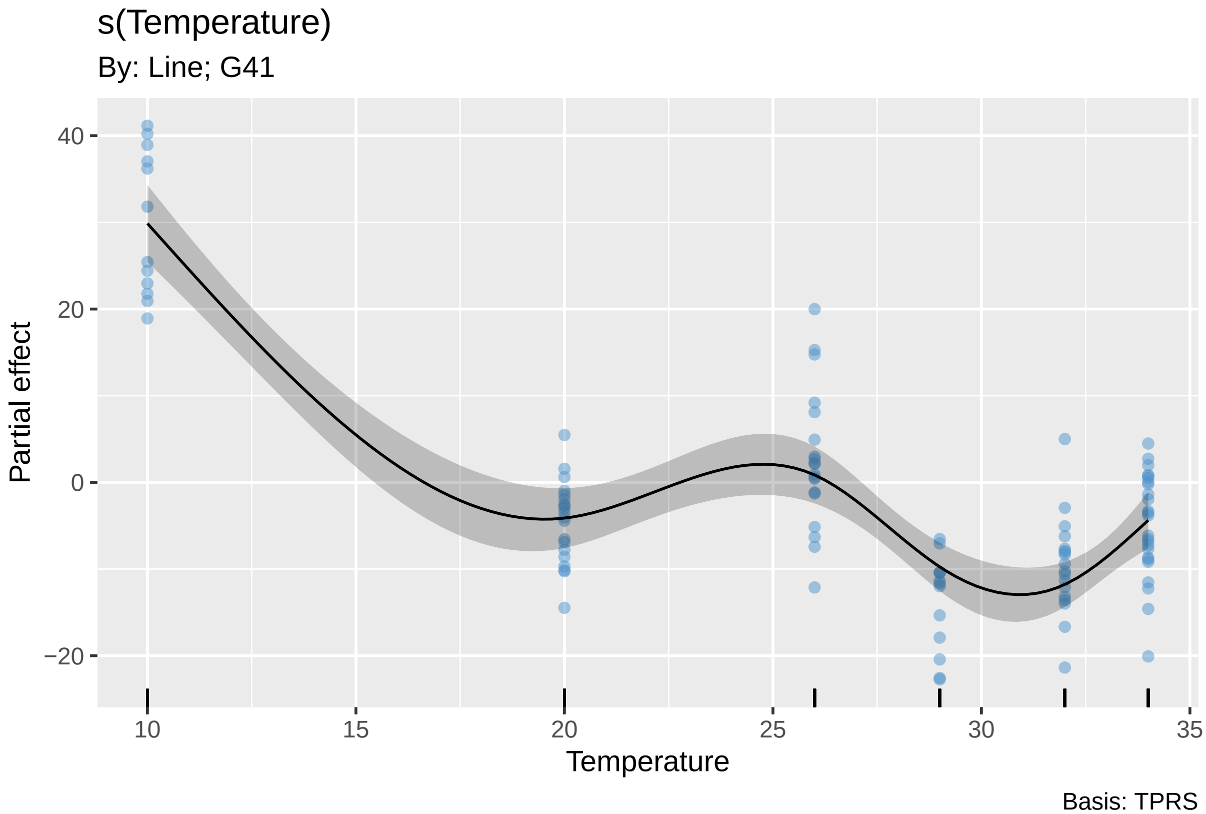

s(Temperature)
By: Line; G33
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

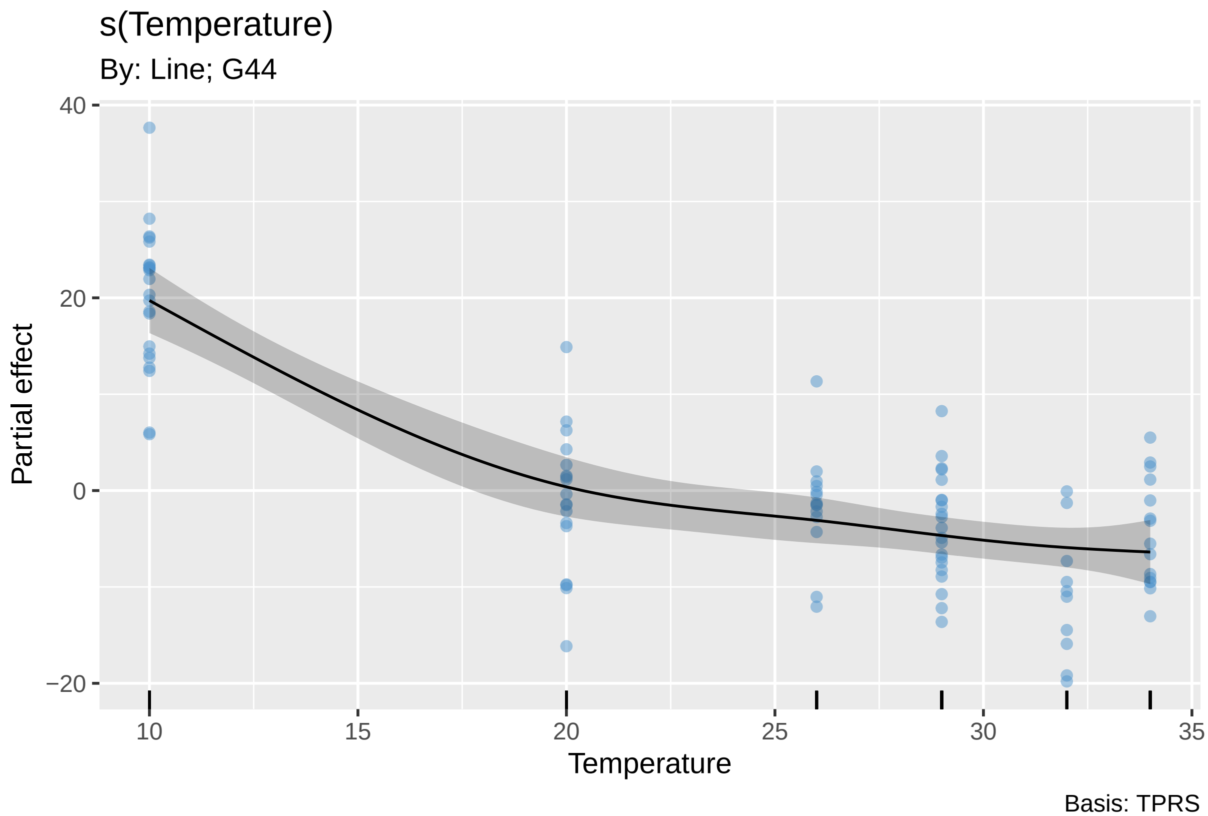

s(Temperature)
By: Line; G34
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


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

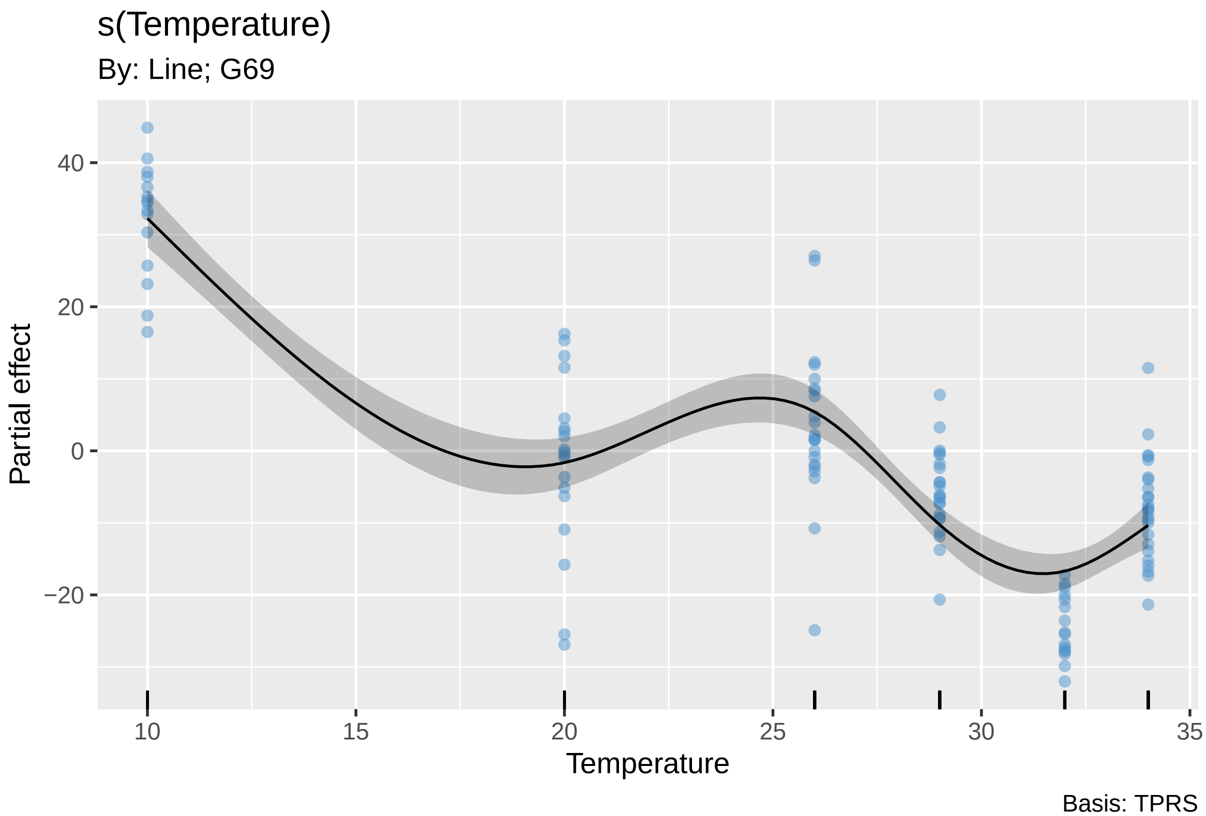

s(Temperature)
By: Line; G36
Partial effect
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

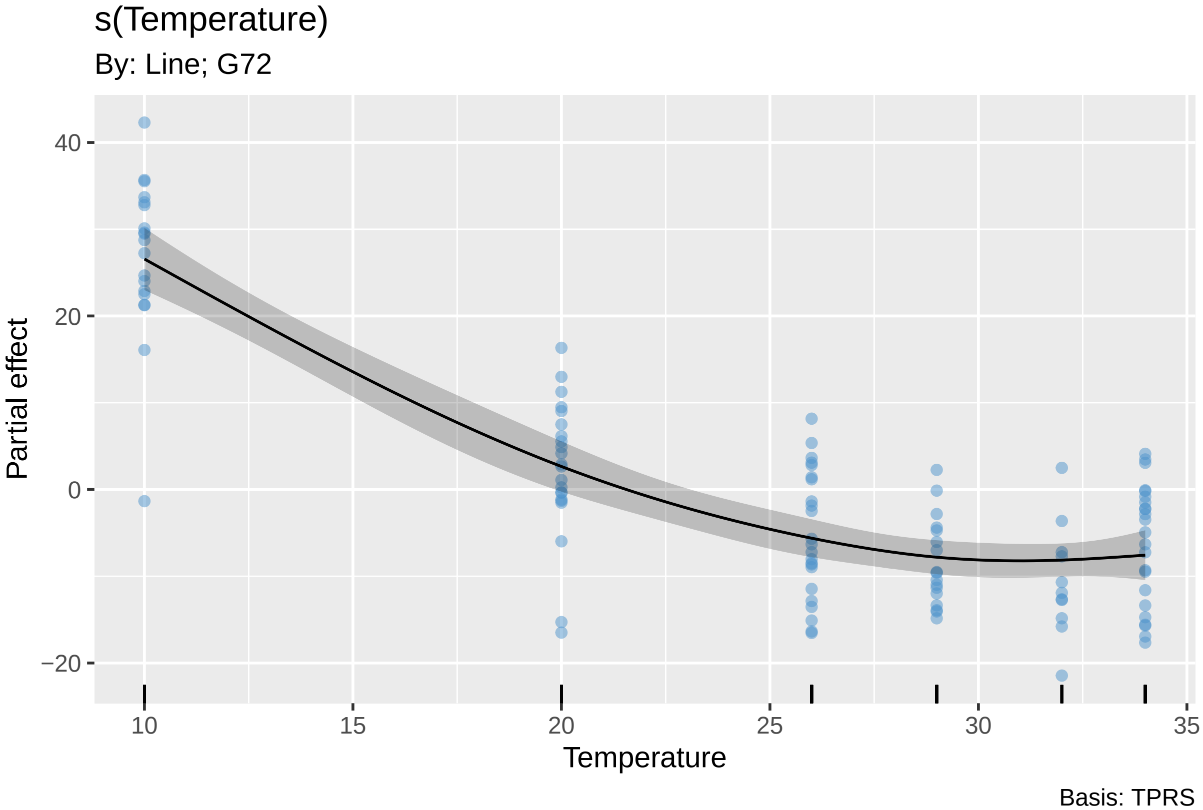

s(Temperature)
By: Line; G38
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

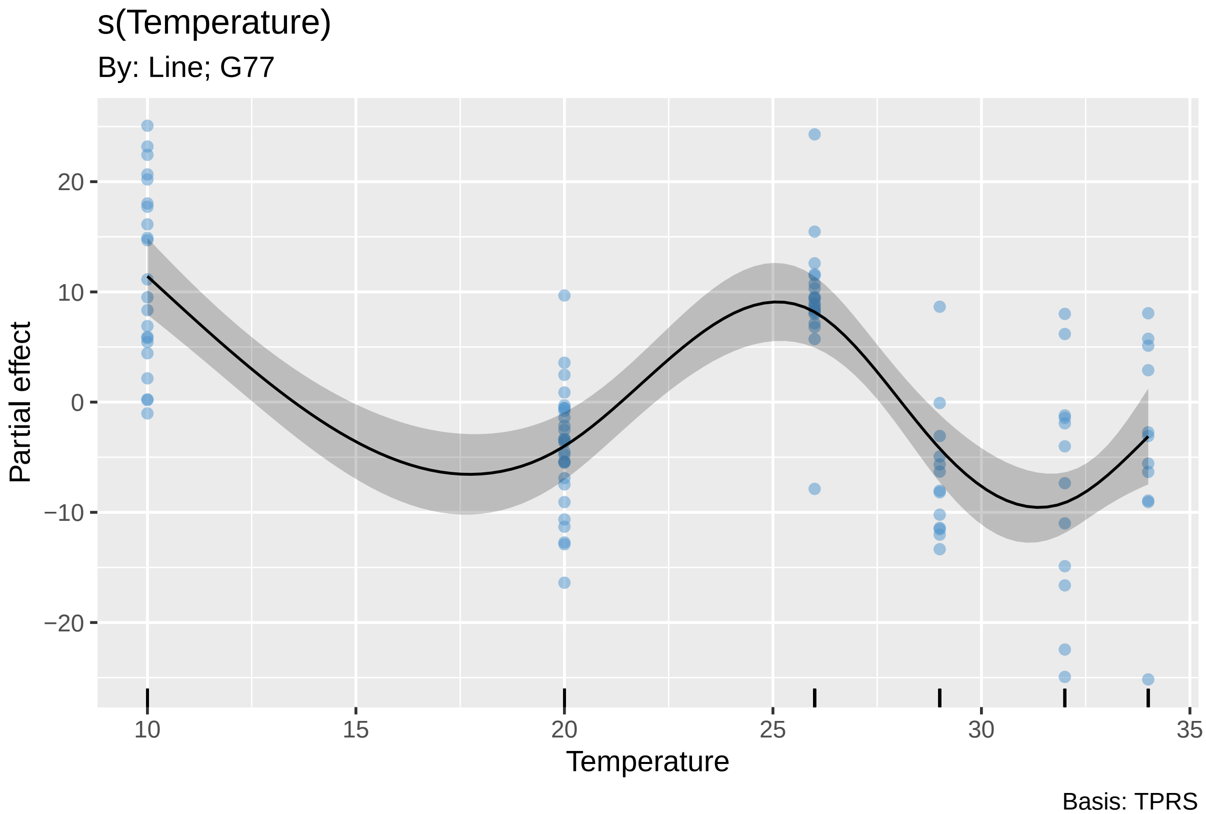

s(Temperature)
By: Line; G41
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS



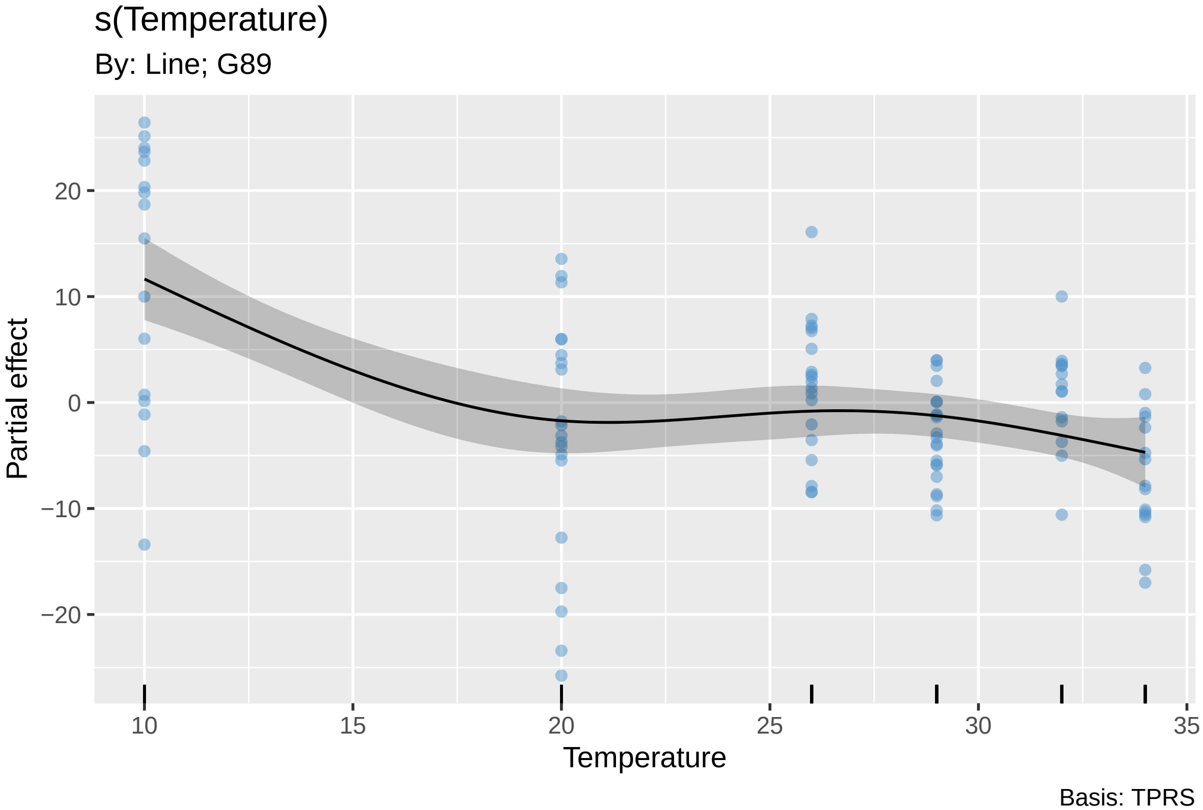
s(Temperature)
By: Line; G44
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS

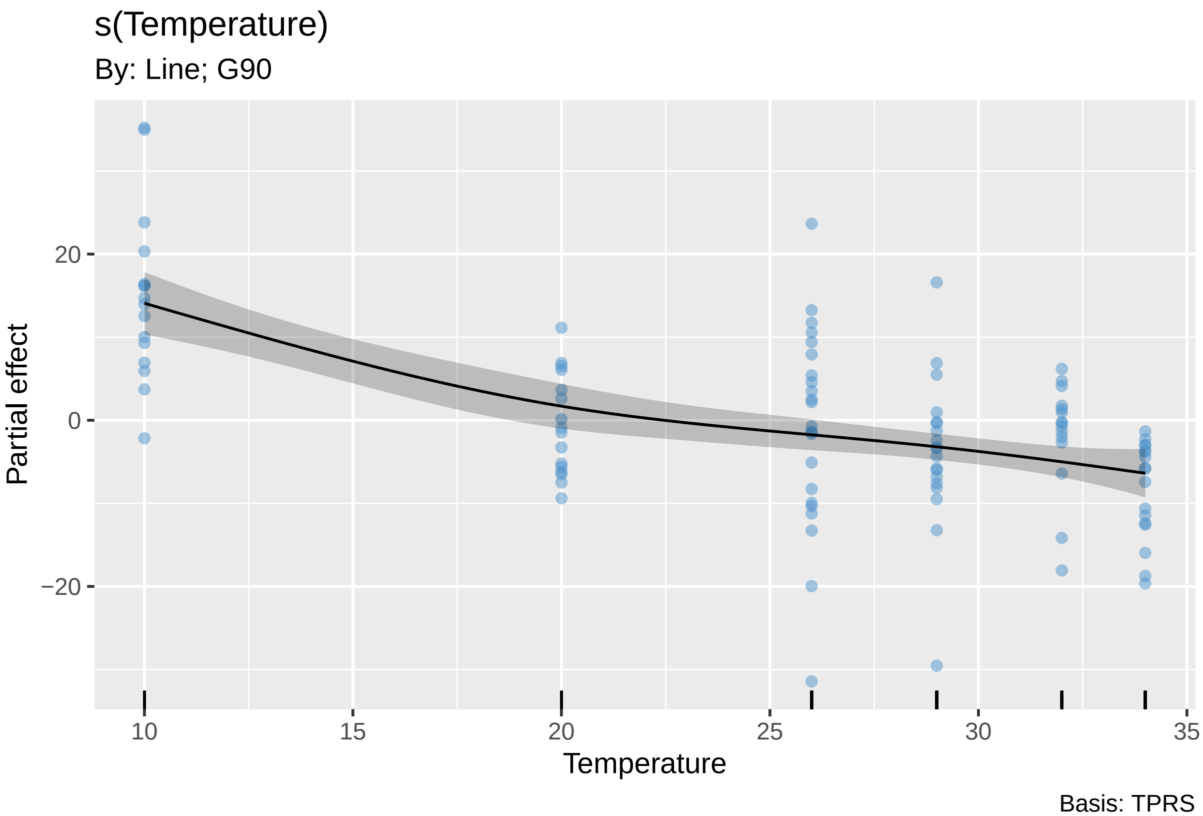

s(Temperature)
By: Line; G69
Partial effect
40
20
0
−20
10
15
20
25
30
35
Temperature
Basis: TPRS


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
20
10
0
−10
−20
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G89
20
10
0
−10
−20
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_width_noint <- gam(formula = mean_minor ~ Line + s(Temperature, k = 5, bs = 'tp'), data = morph_data

AIC(gam_width, gam_width_noint)
```

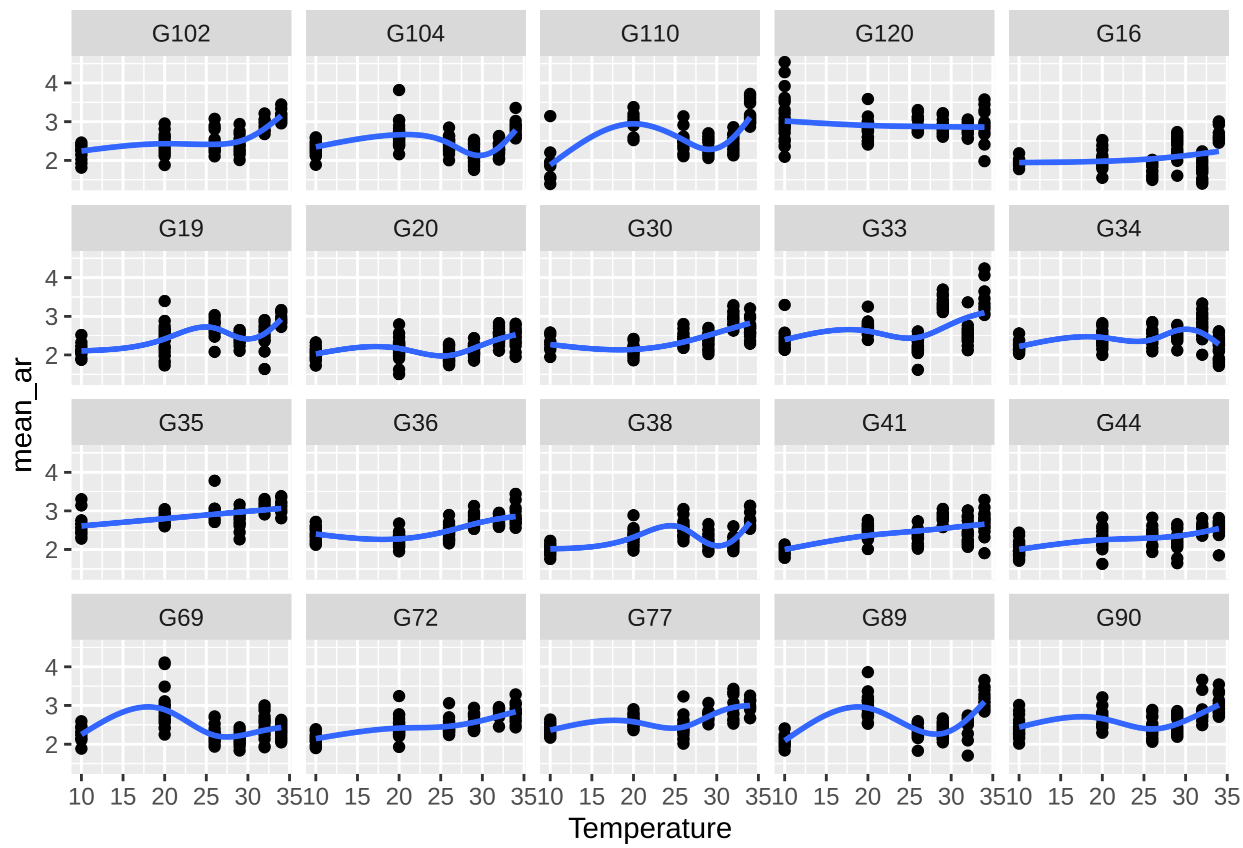
```
##                       df      AIC
## gam_width       77.27334 15133.30
## gam_width_noint 24.99329 15598.34
```

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)
```



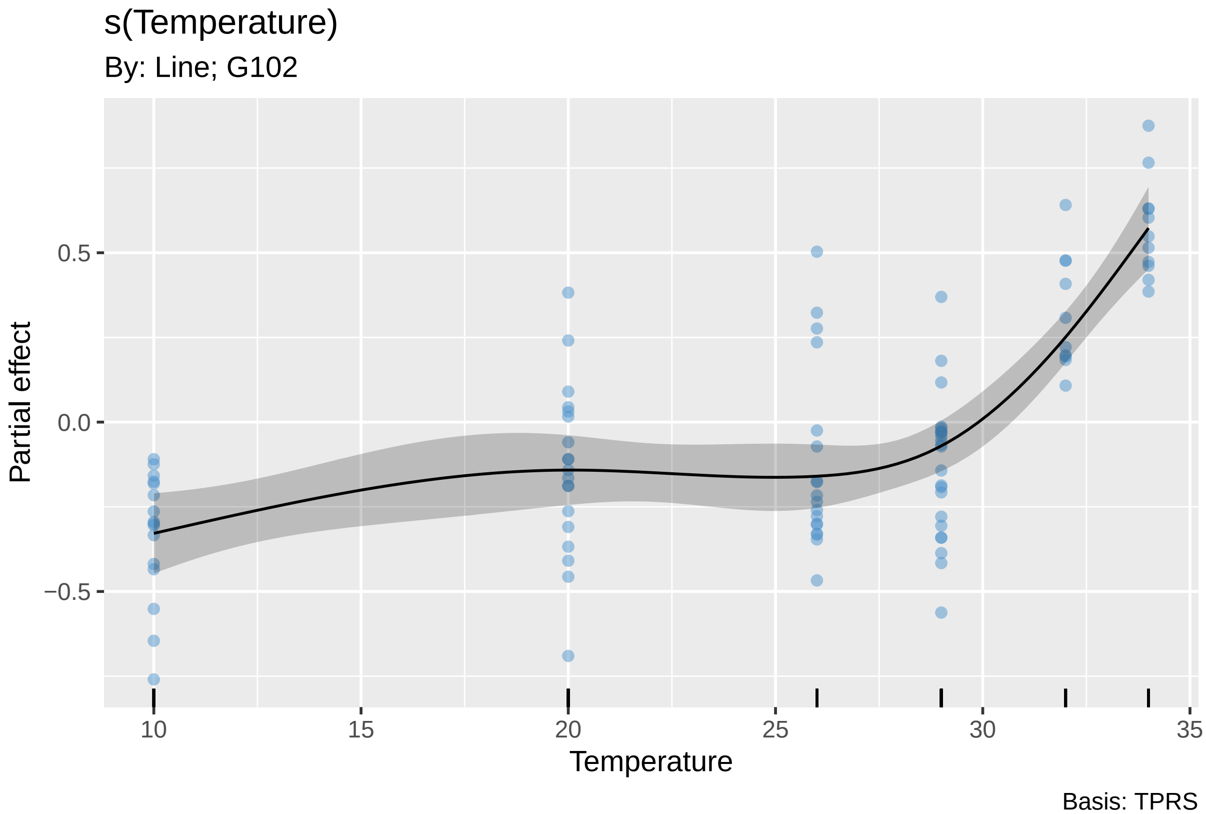
Overall, across the different lines, the general trend is for an overall increase in aspect ratio with temperature. Fitting a GAM gives:

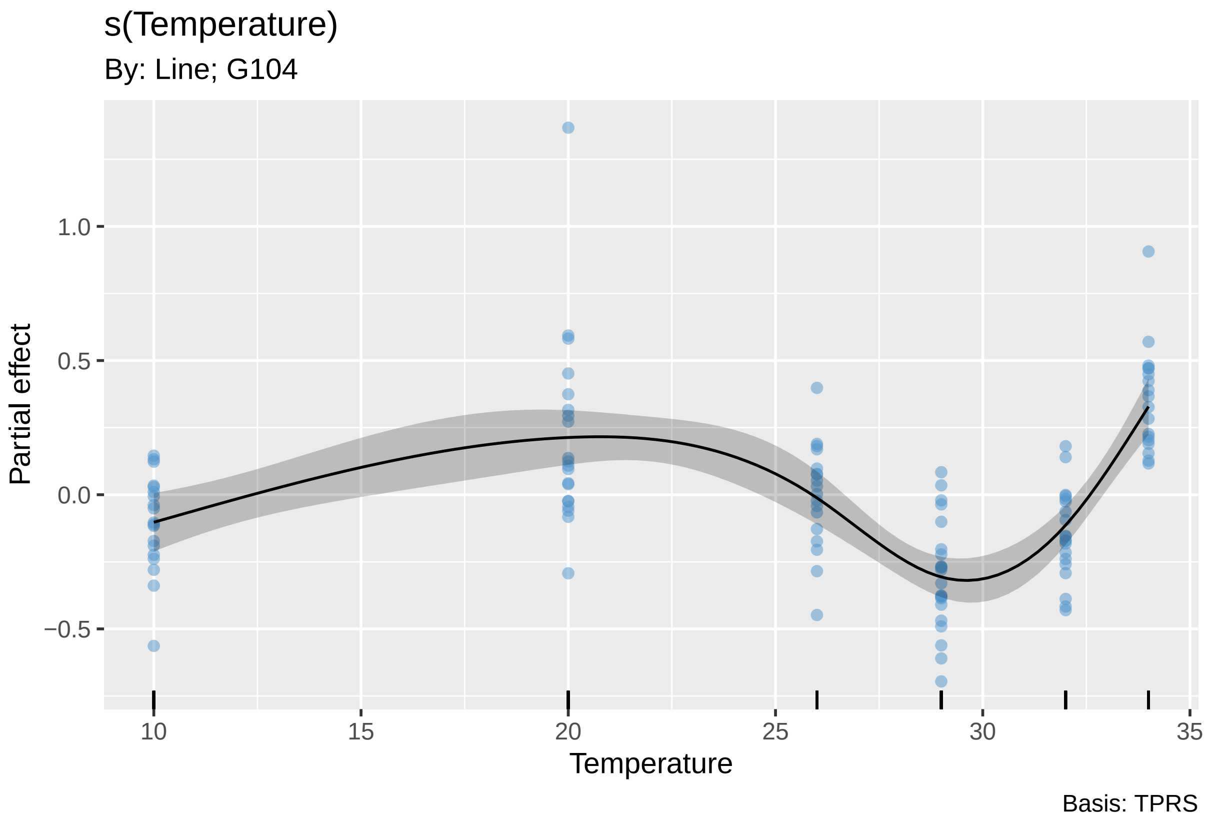
```
gam_ar <- gam(formula = mean_ar ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = morph_data

summary(gam_ar)
```

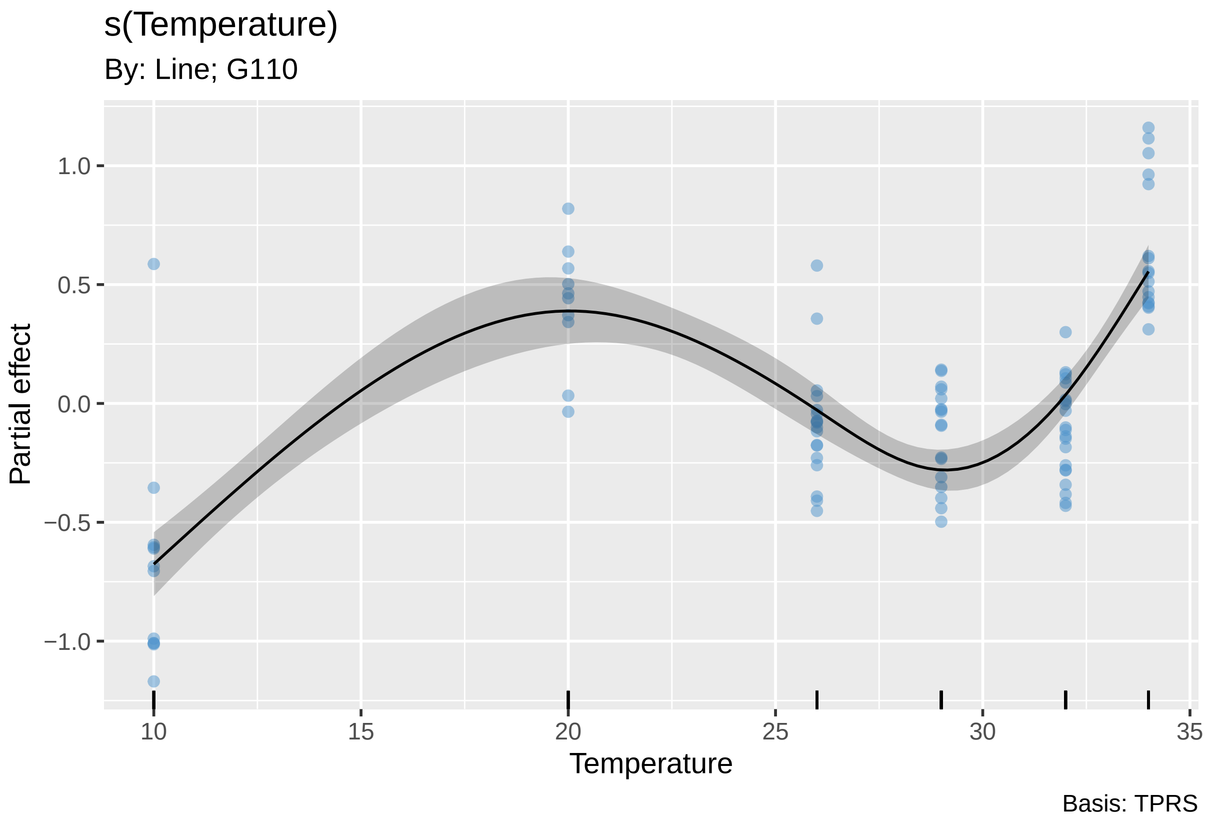
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## mean_ar ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.570162   0.026828  95.801  < 2e-16 ***
## LineG104    -0.120570   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 ***
```

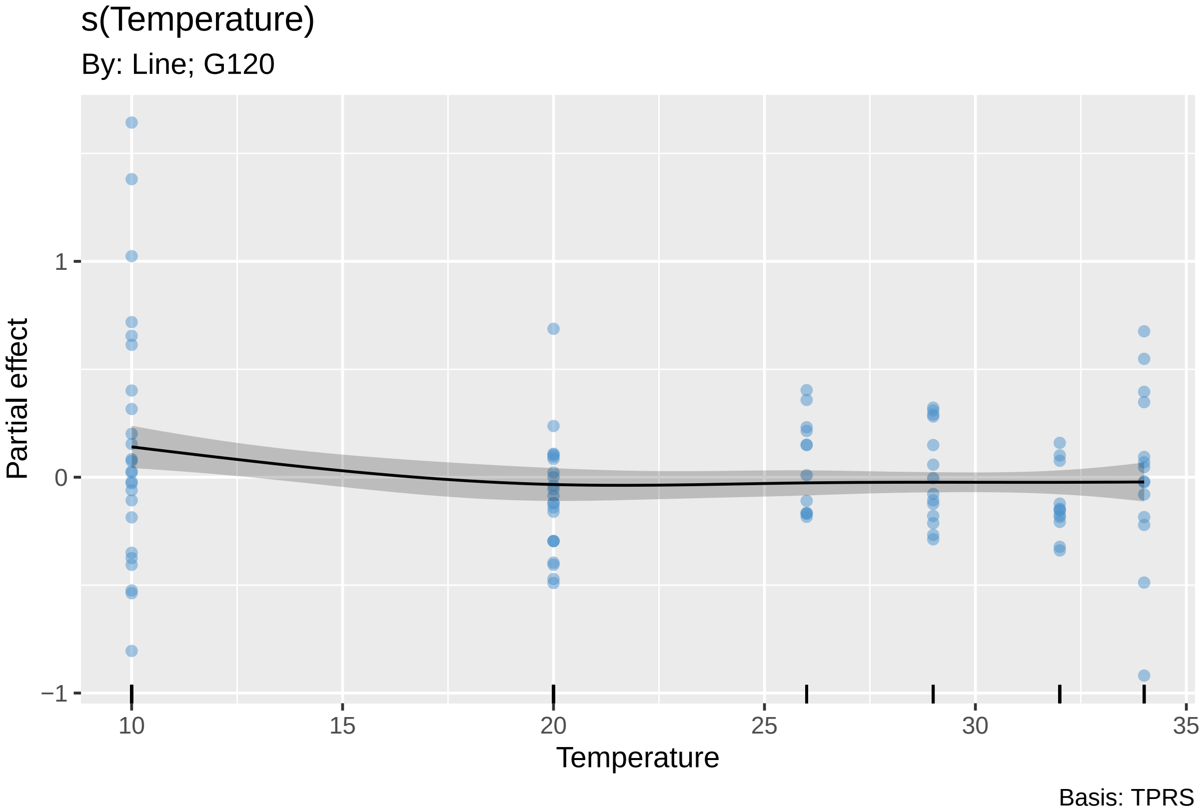
```
## LineG36     -0.007736   0.038614  -0.200 0.841242
## LineG38     -0.240060   0.036317  -6.610 4.90e-11 ***
## LineG41     -0.130129   0.037200  -3.498 0.000479 ***
## LineG44     -0.259106   0.037517  -6.906 6.64e-12 ***
## LineG69     -0.164042   0.036184  -4.533 6.14e-06 ***
## LineG72     -0.054556   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
```

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

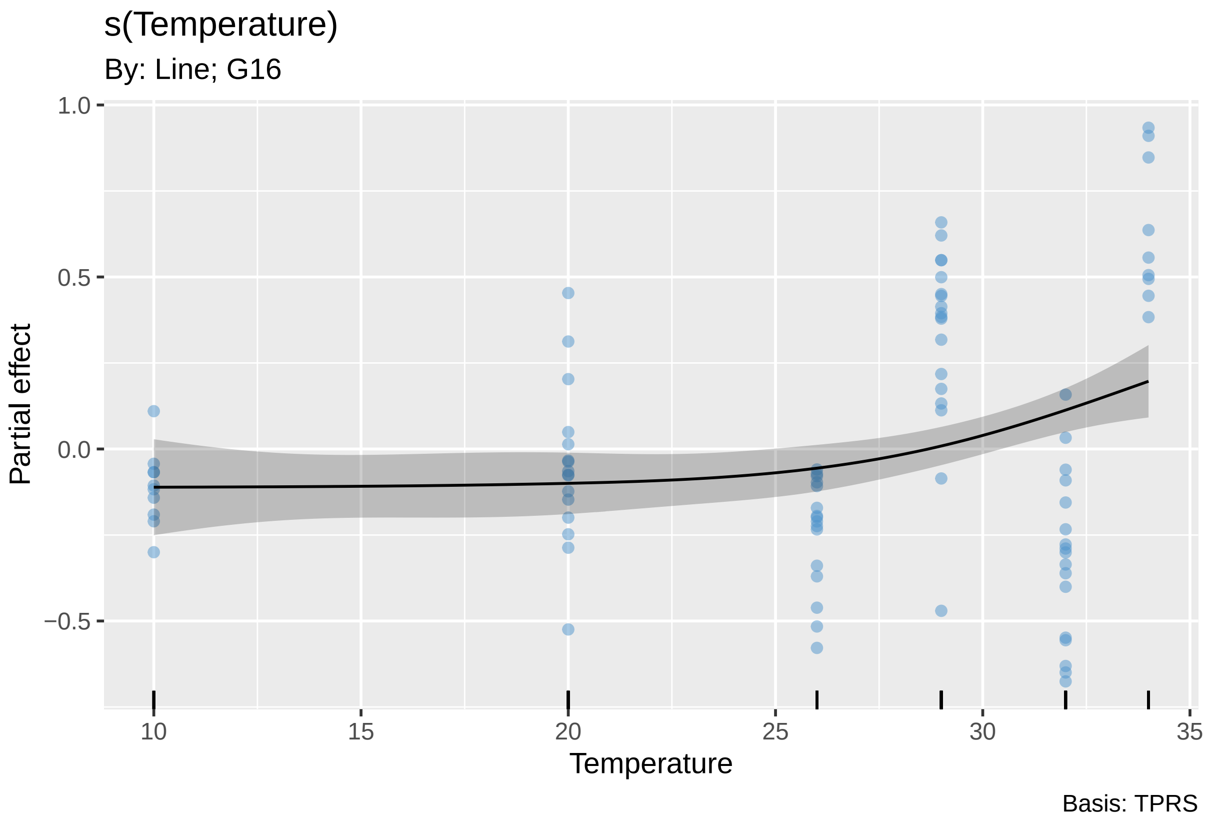

s(Temperature)
By: Line; G102
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

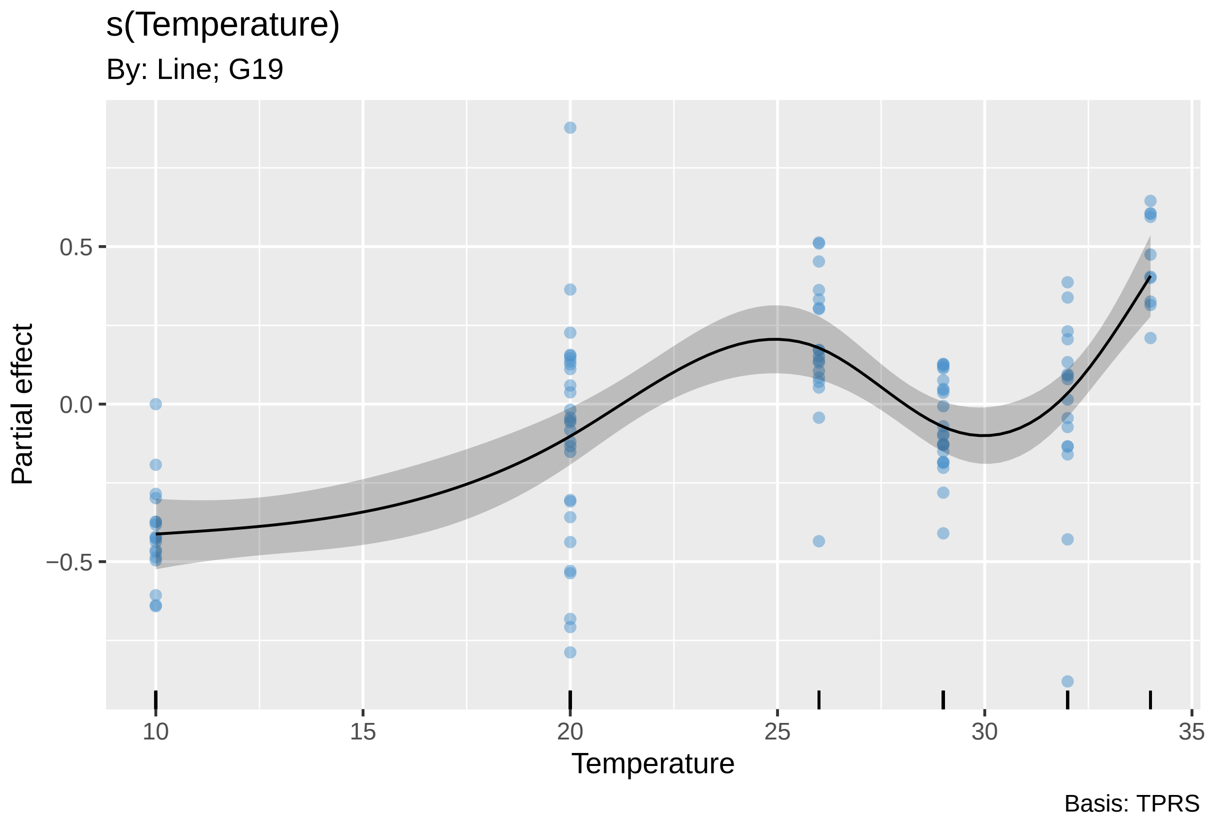

s(Temperature)
By: Line; G104
Partial effect
1.0
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

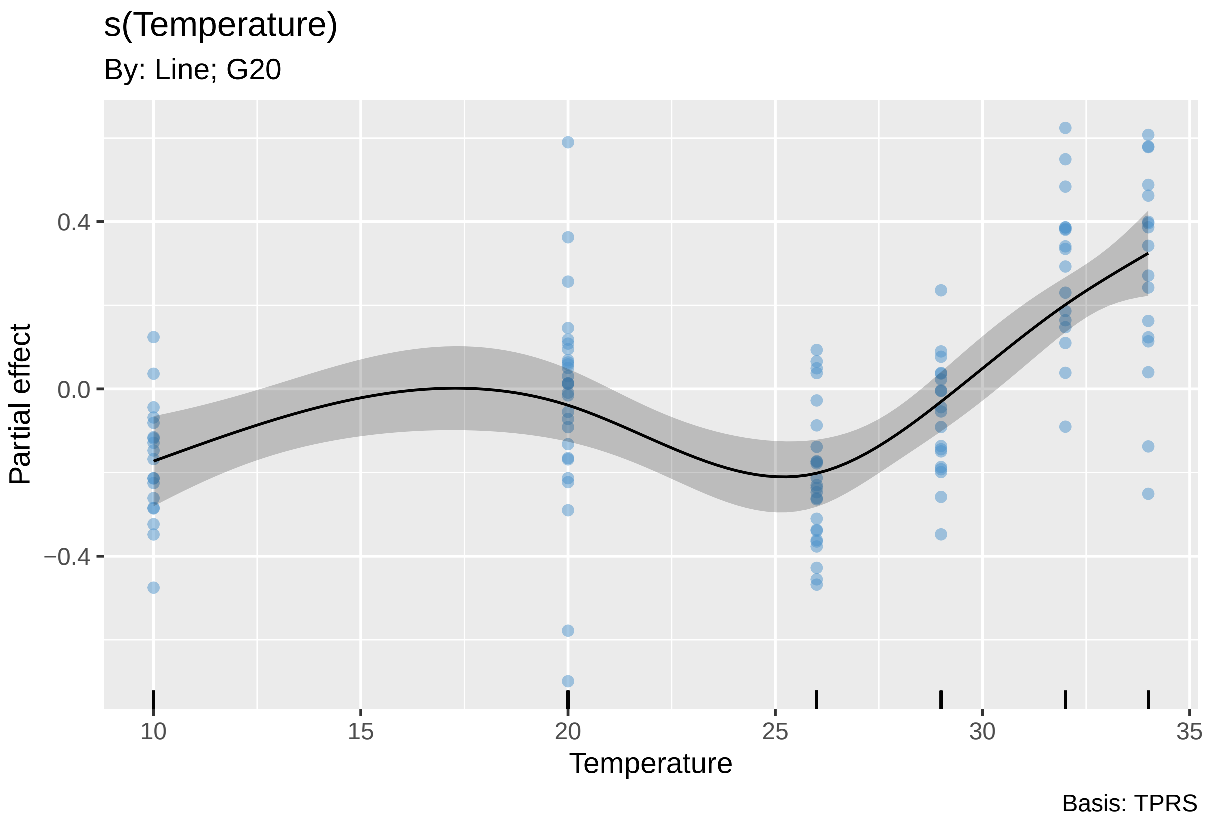

s(Temperature)
By: Line; G110
Partial effect
1.0
0.5
0.0
−0.5
−1.0
10
15
20
25
30
35
Temperature
Basis: TPRS

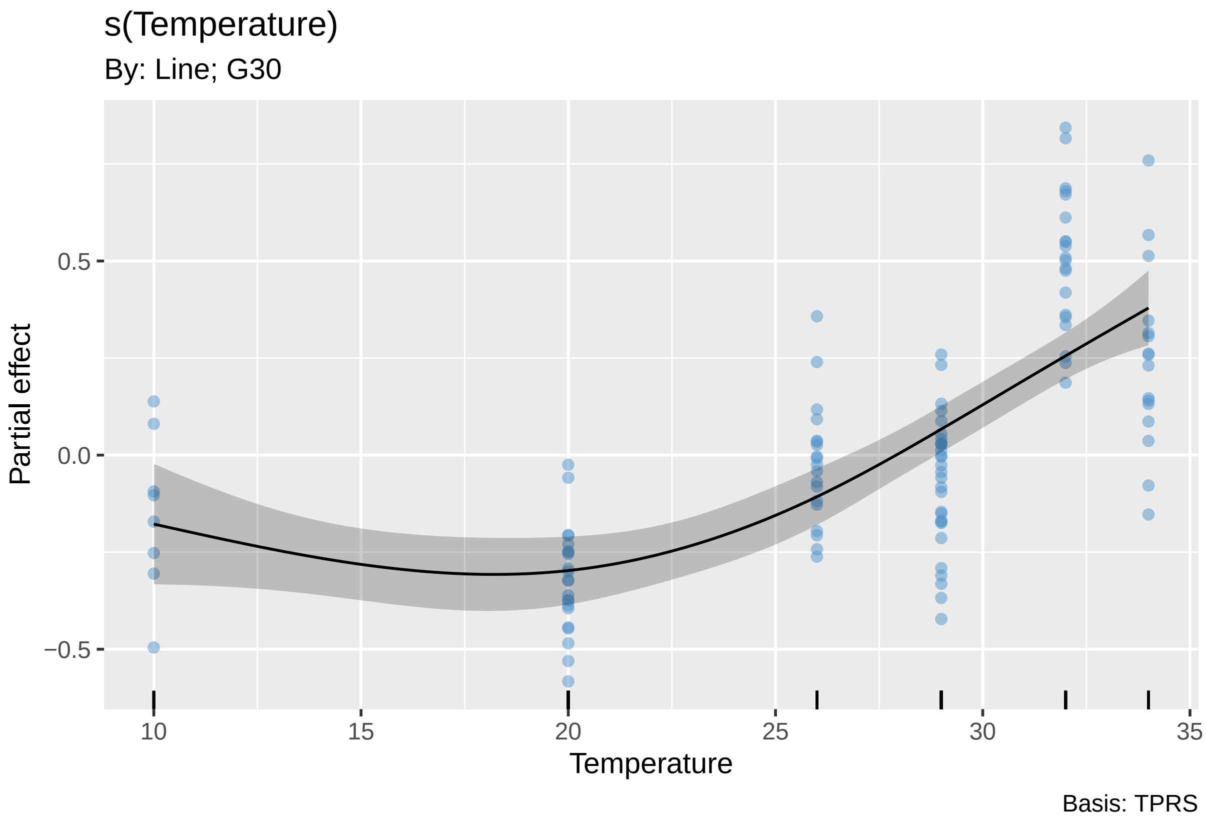


s(Temperature)
By: Line; G120
Partial effect
1
0
−1
10
15
20
25
30
35
Temperature
Basis: TPRS

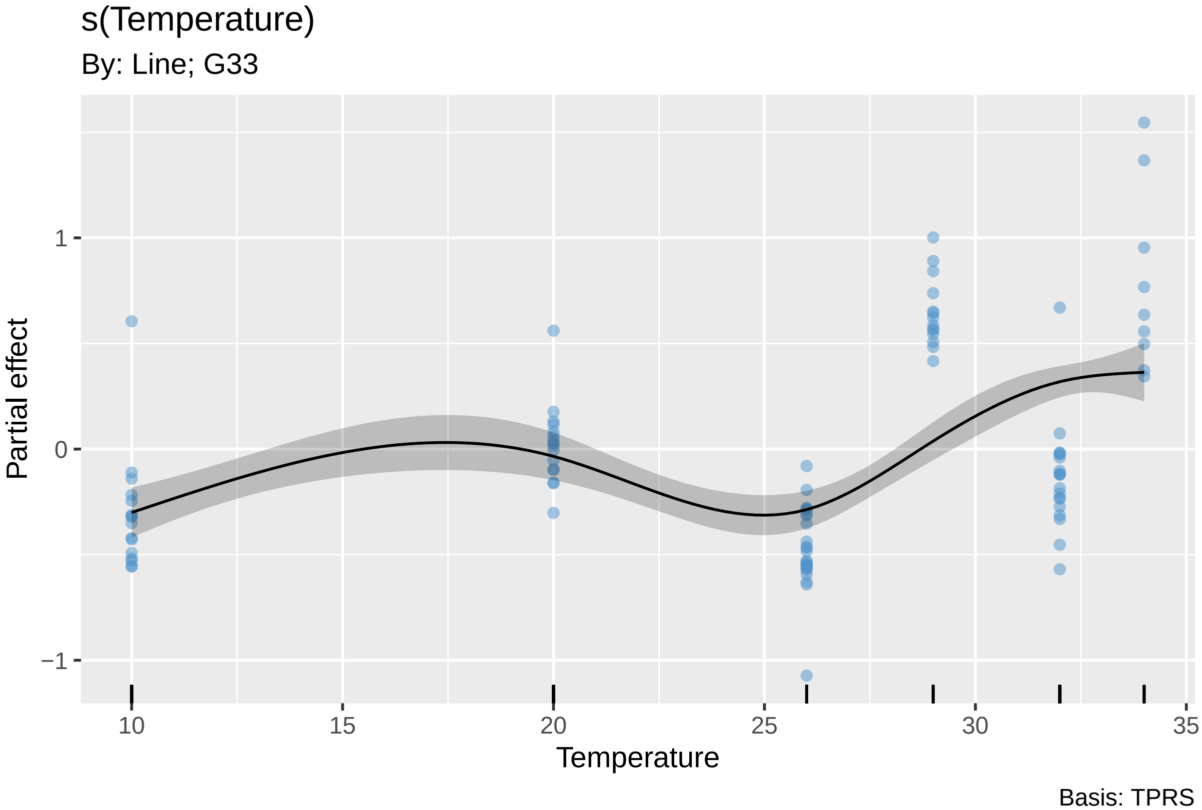

s(Temperature)
By: Line; G16
1.0
0.5
0.0
−0.5
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

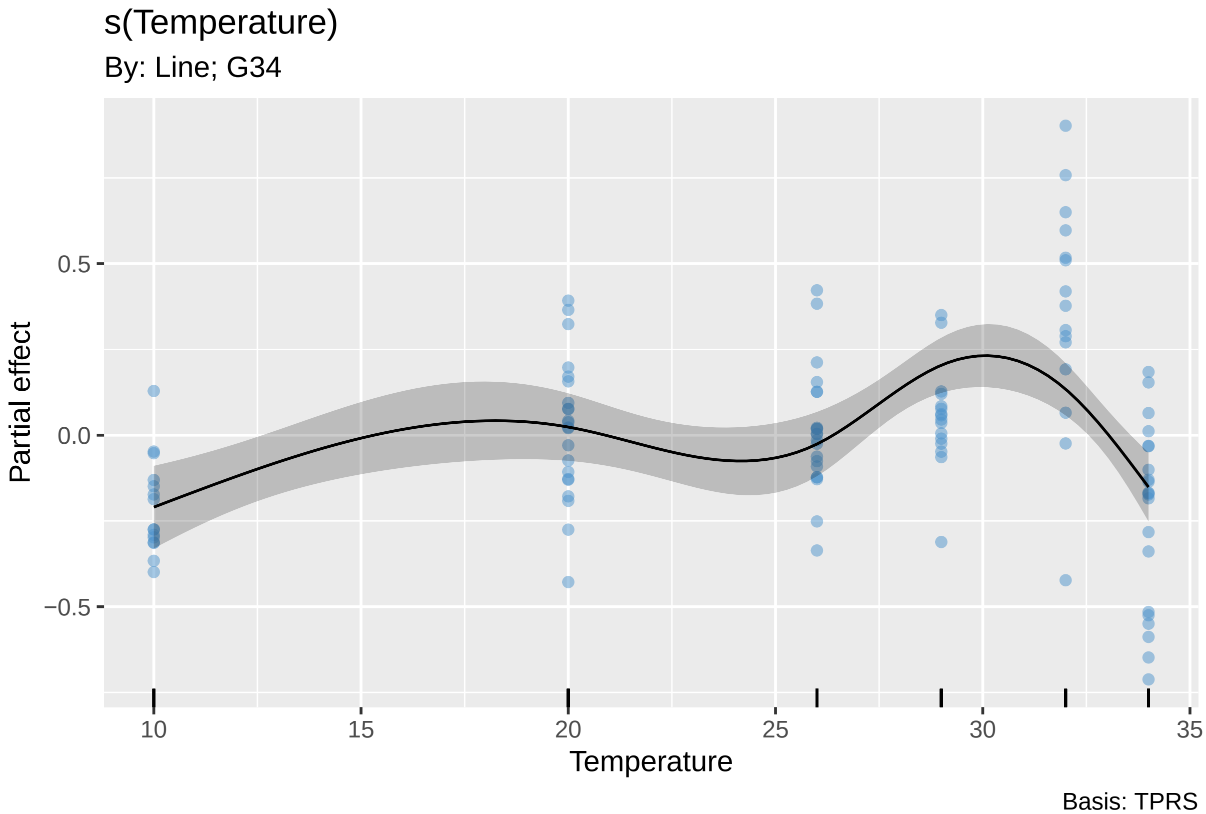

s(Temperature)
By: Line; G19
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS


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

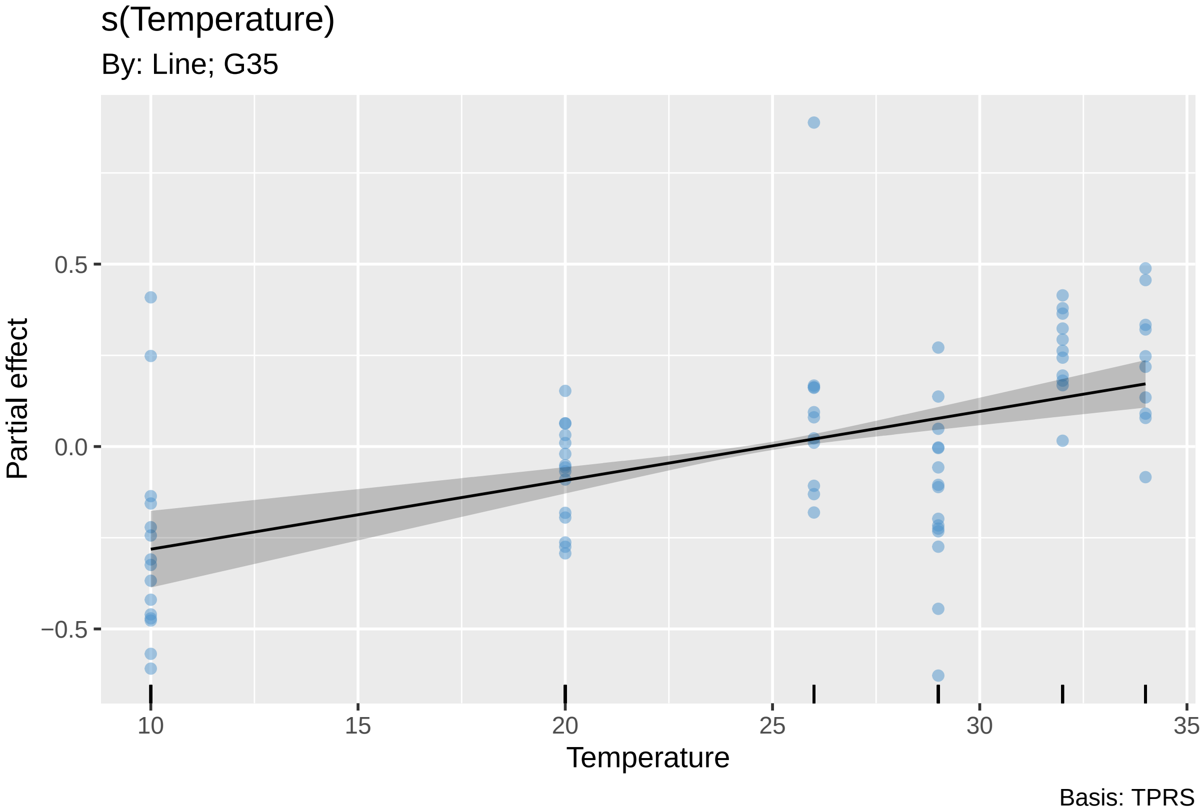


s(Temperature)
By: Line; G30
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

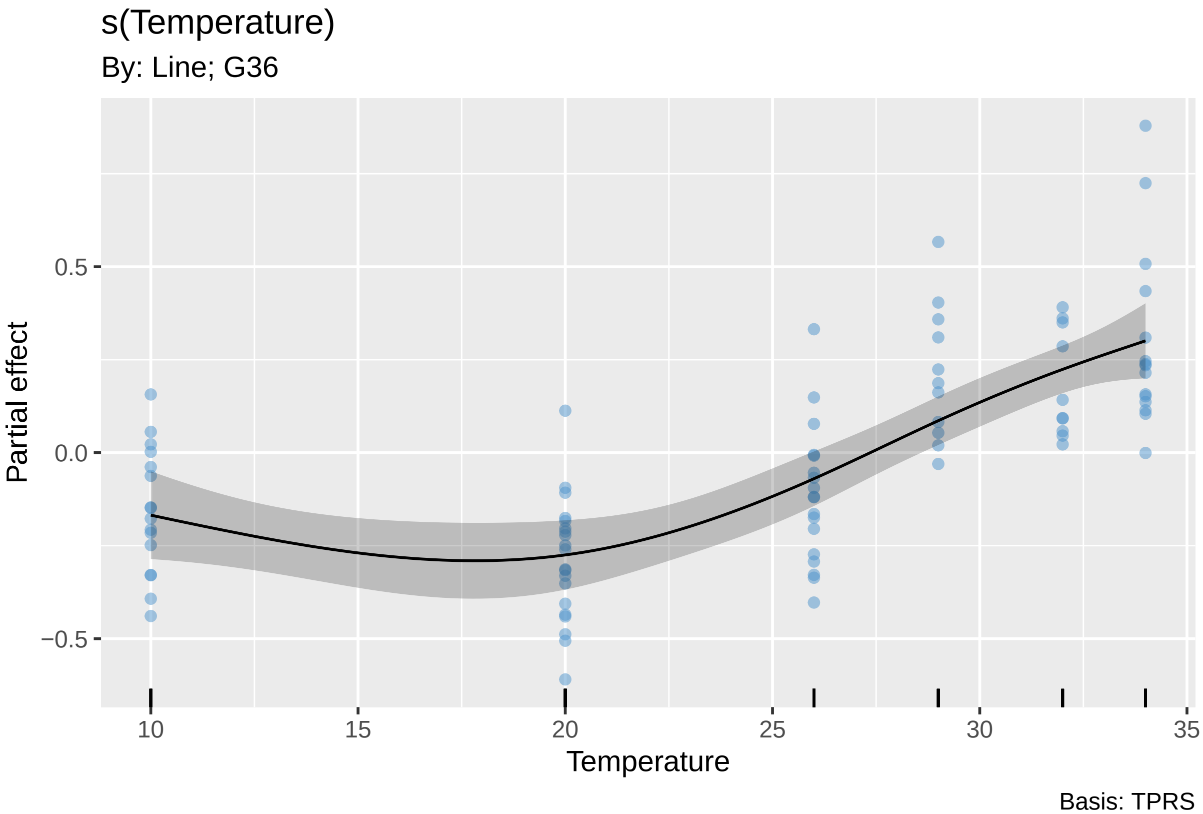

s(Temperature)
By: Line; G33
Partial effect
1
0
−1
10
15
20
25
30
35
Temperature
Basis: TPRS



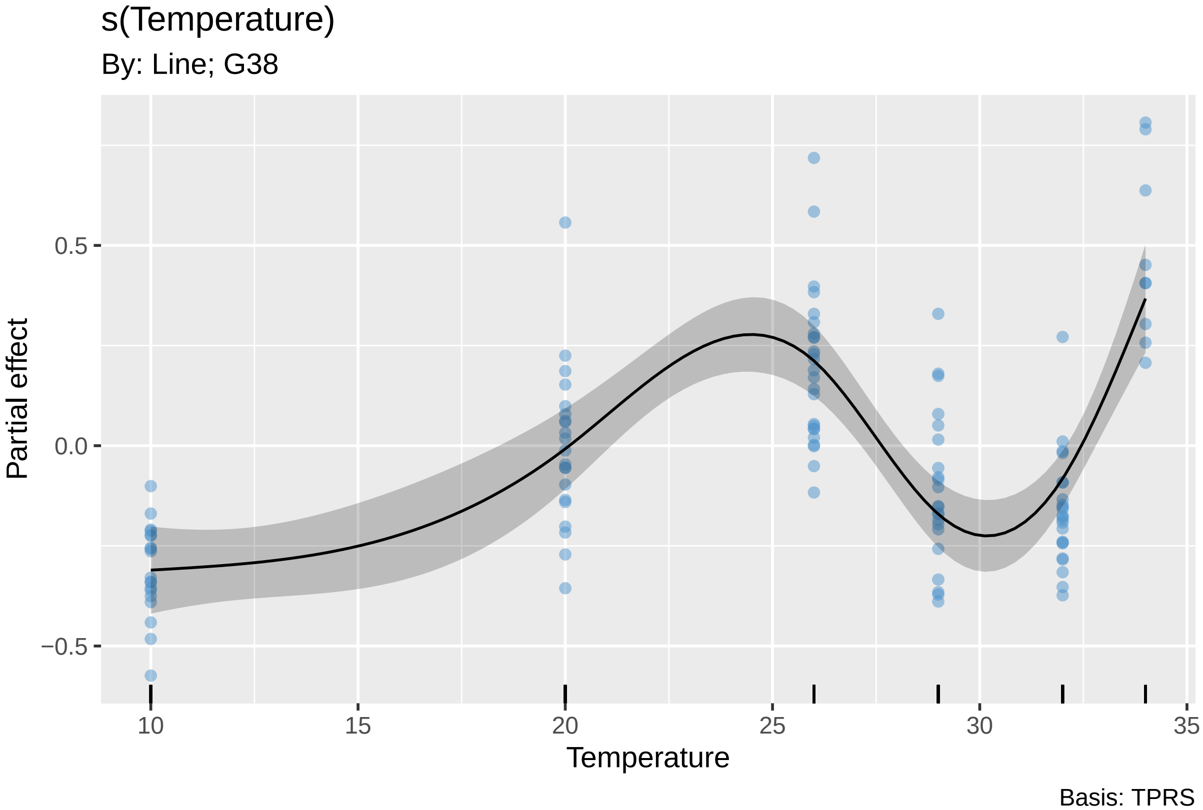
s(Temperature)
By: Line; G34
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS

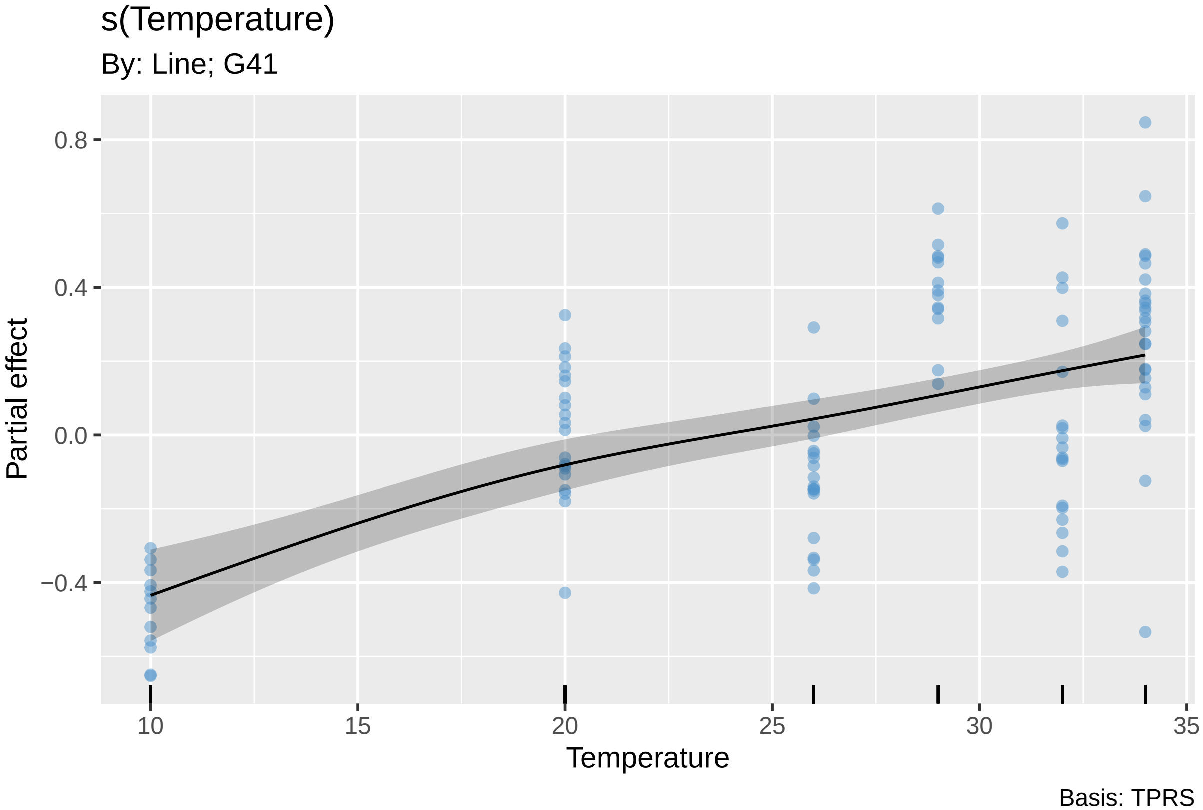


s(Temperature)
By: Line; G35
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS



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

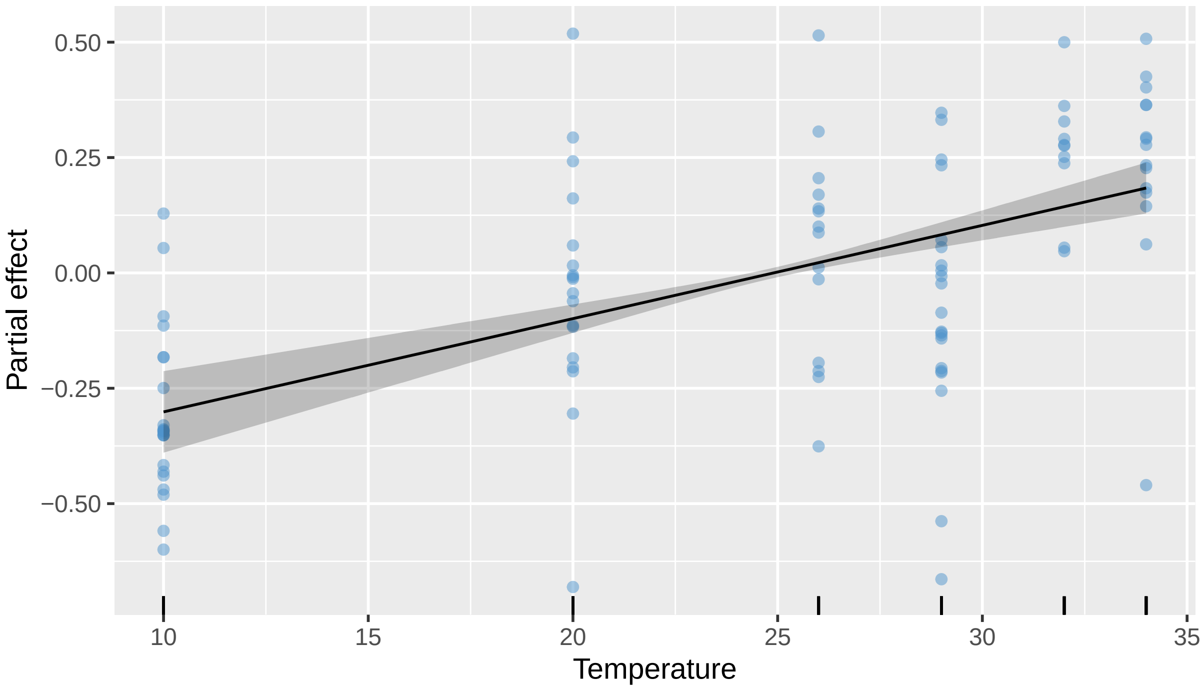

s(Temperature)
By: Line; G38
Partial effect
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS



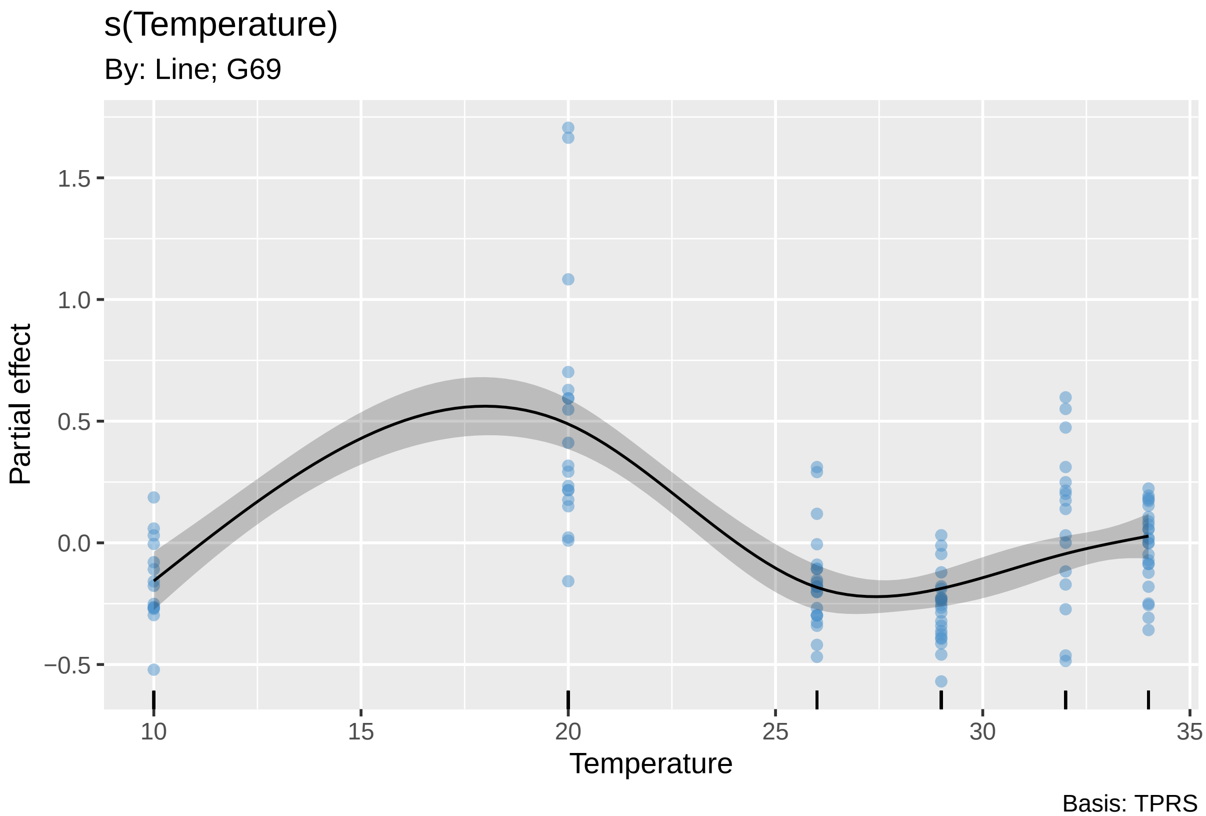
s(Temperature)
By: Line; G41
Partial effect
0.8
0.4
0.0
−0.4
10
15
20
25
30
35
Temperature
Basis: TPRS

s(Temperature)

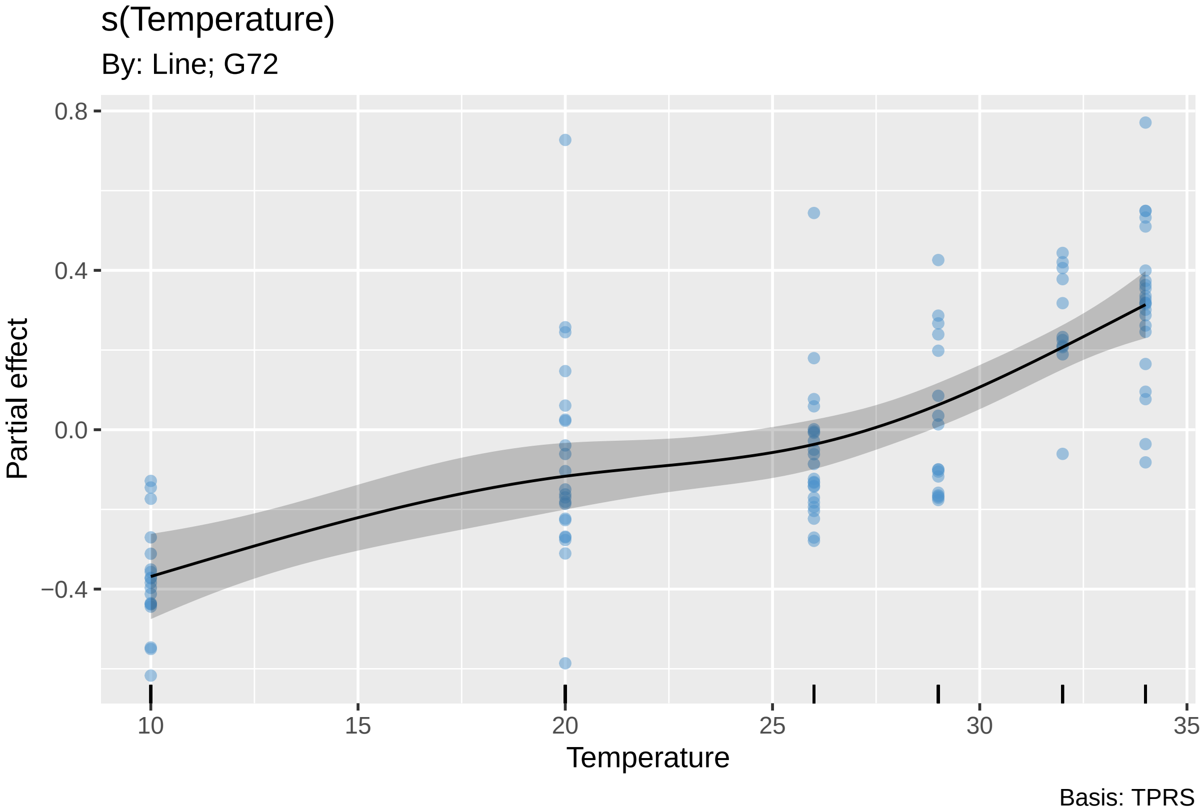
By: Line; G44

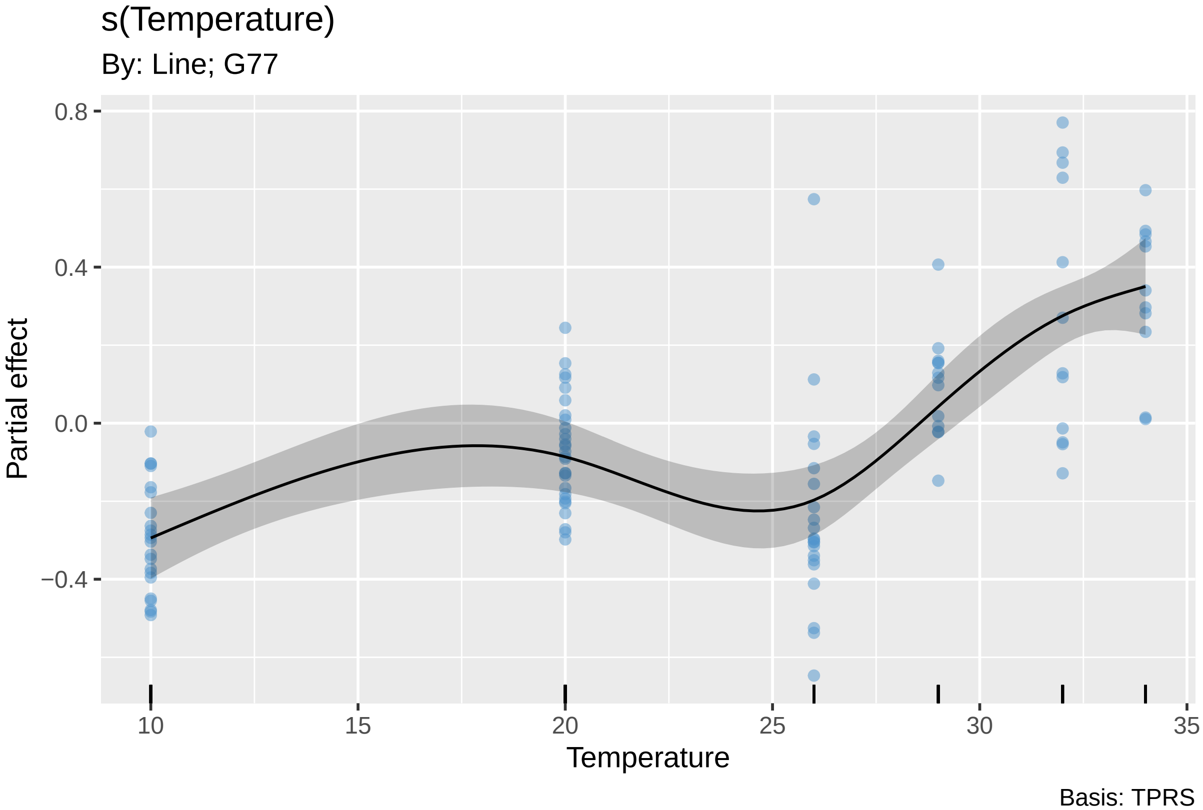


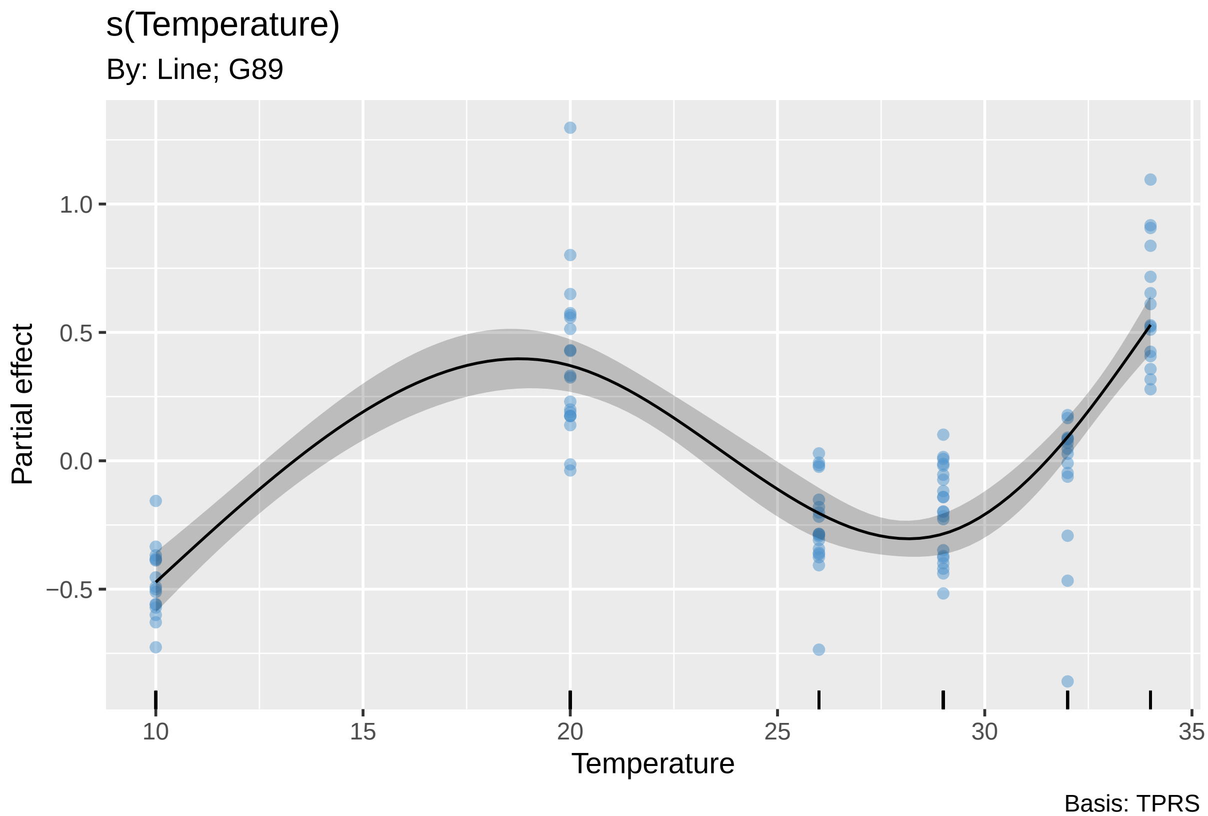
Basis: TPRS

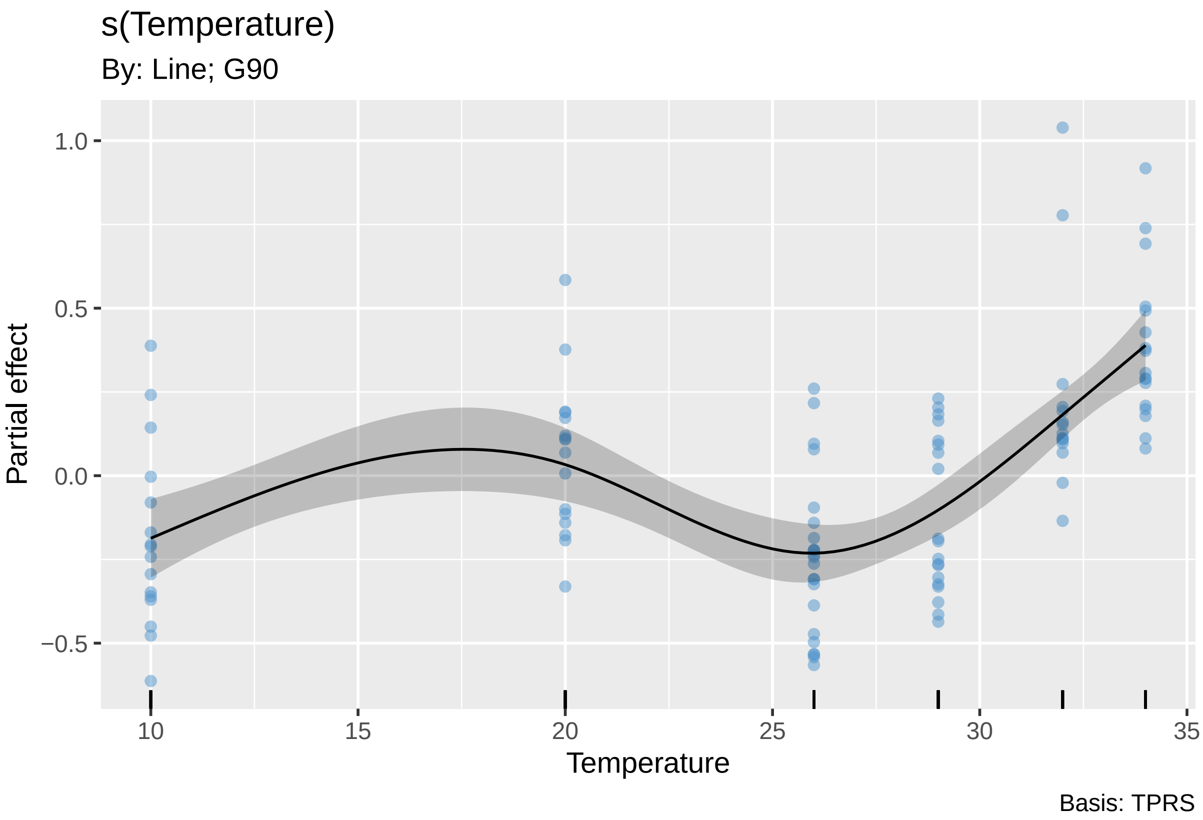


s(Temperature)
By: Line; G69
Partial effect
1.5
1.0
0.5
0.0
−0.5
10
15
20
25
30
35
Temperature
Basis: TPRS


s(Temperature)
By: Line; G72
0.8
0.4
0.0
−0.4
Partial effect
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)
```

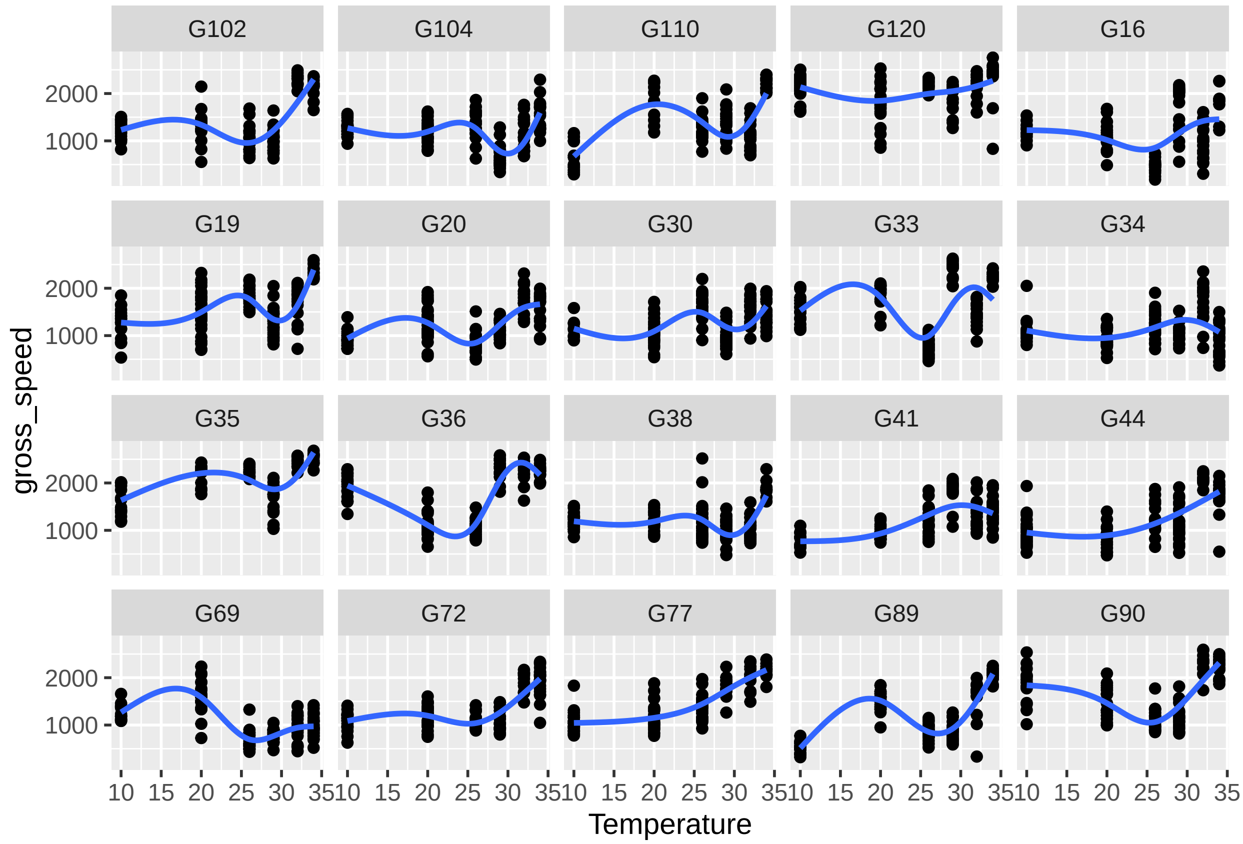
```
##                    df       AIC
## gam_ar       87.50686  409.7618
## gam_ar_noint 24.91808 1010.1726
```

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)
```



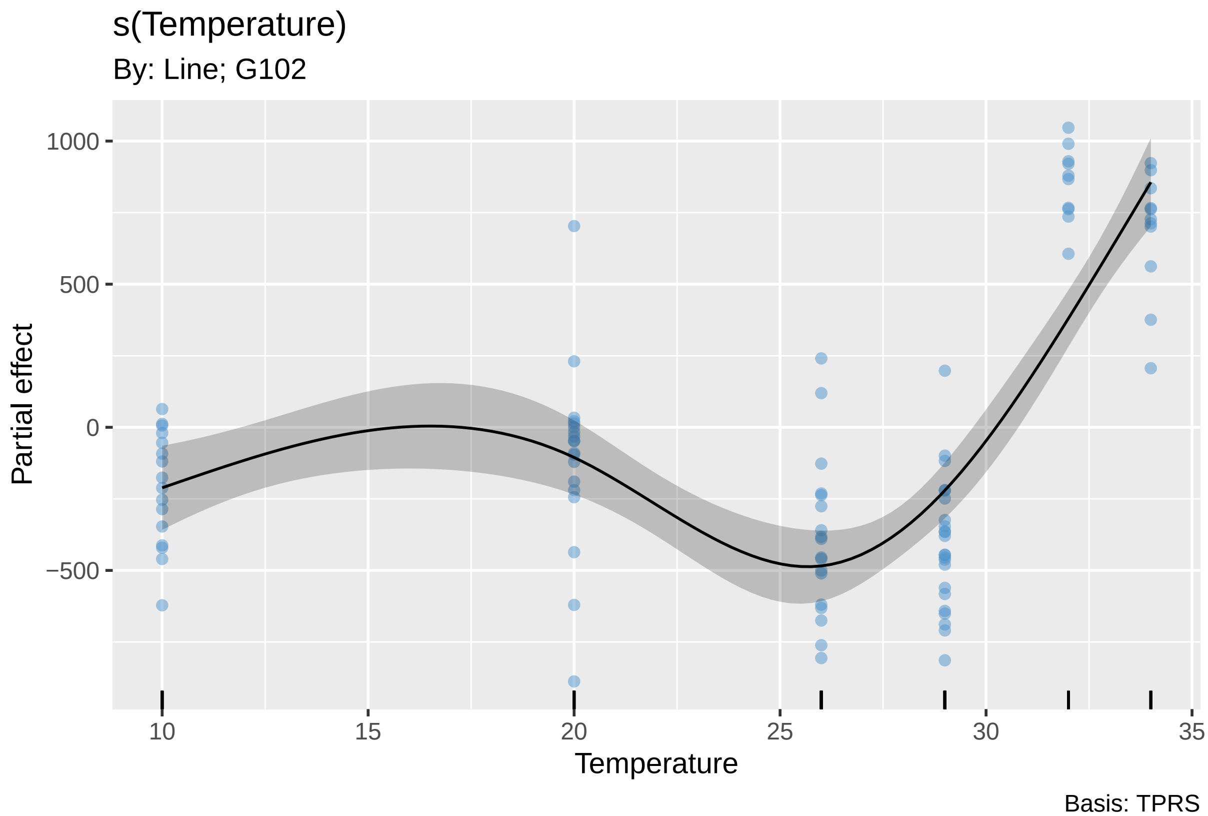
As for aspect ratio, we some general trends towards faster paramecium at higher temperatures with some exceptions. Again, we can fit a GAM with an interaction between temperature and line.

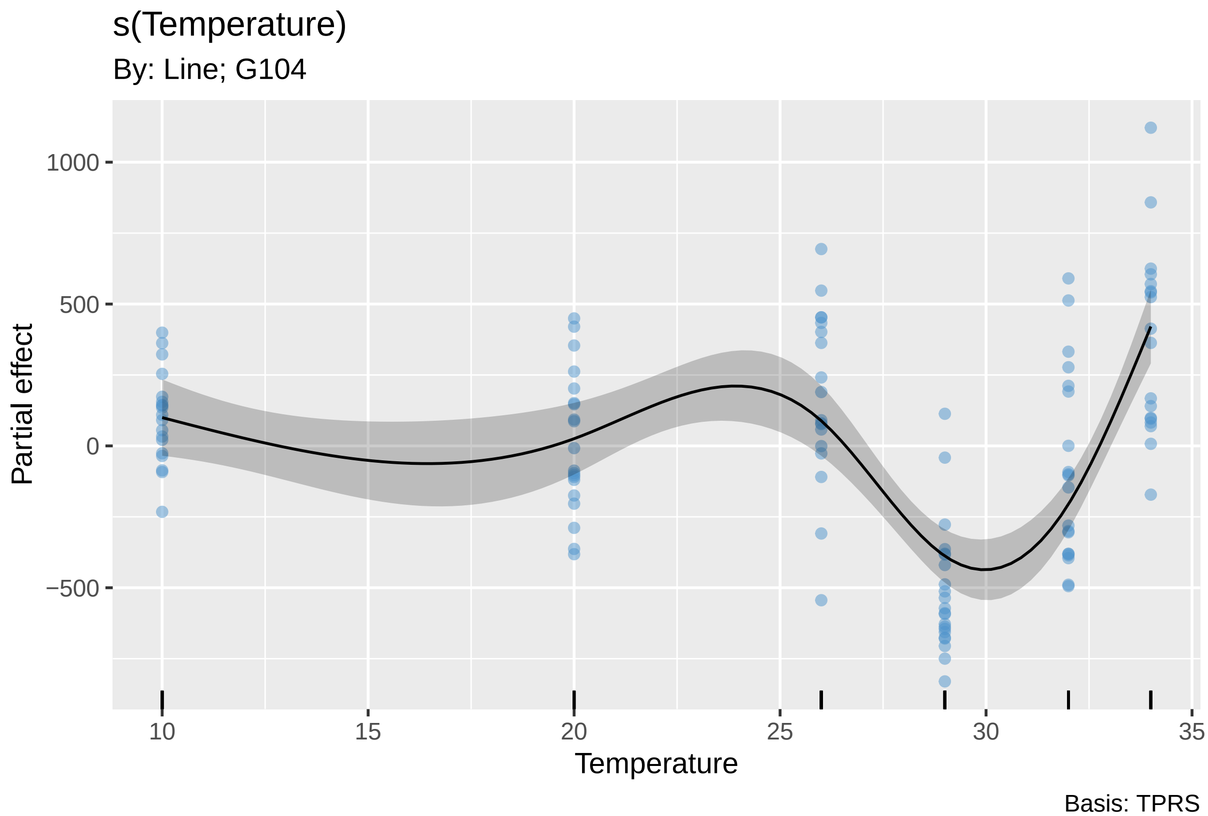
```
gam_speed <- gam(formula = gross_speed ~ Line + s(Temperature, by = Line, k = 5, bs = 'tp'), data = mor

summary(gam_speed)
```

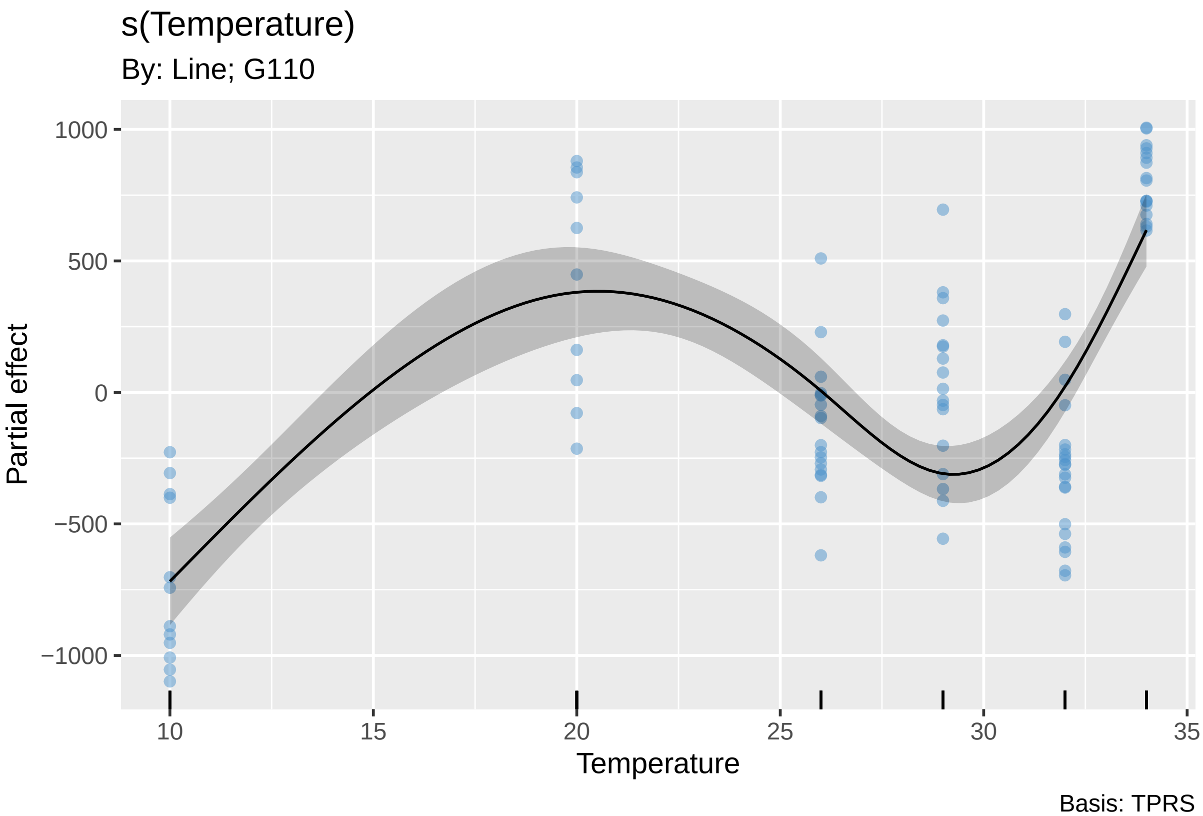
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## gross_speed ~ Line + s(Temperature, by = Line, k = 5, bs = "tp")
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1442.94      33.24  43.407  < 2e-16 ***
## LineG104     -272.12      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
```

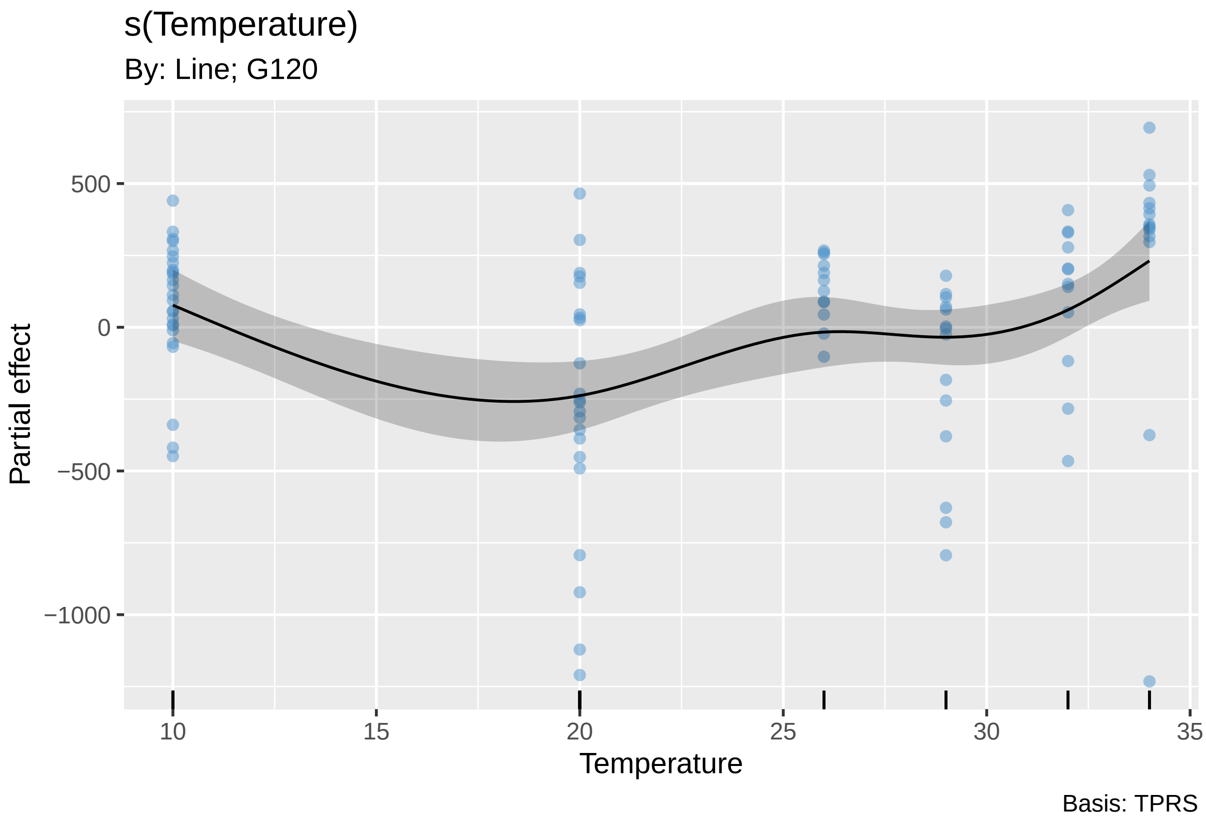
```r
for (i in 1:length(smooths(gam_speed))) {
  out <- draw(gam_speed, select = i, residuals = TRUE)
  print(out) }
```

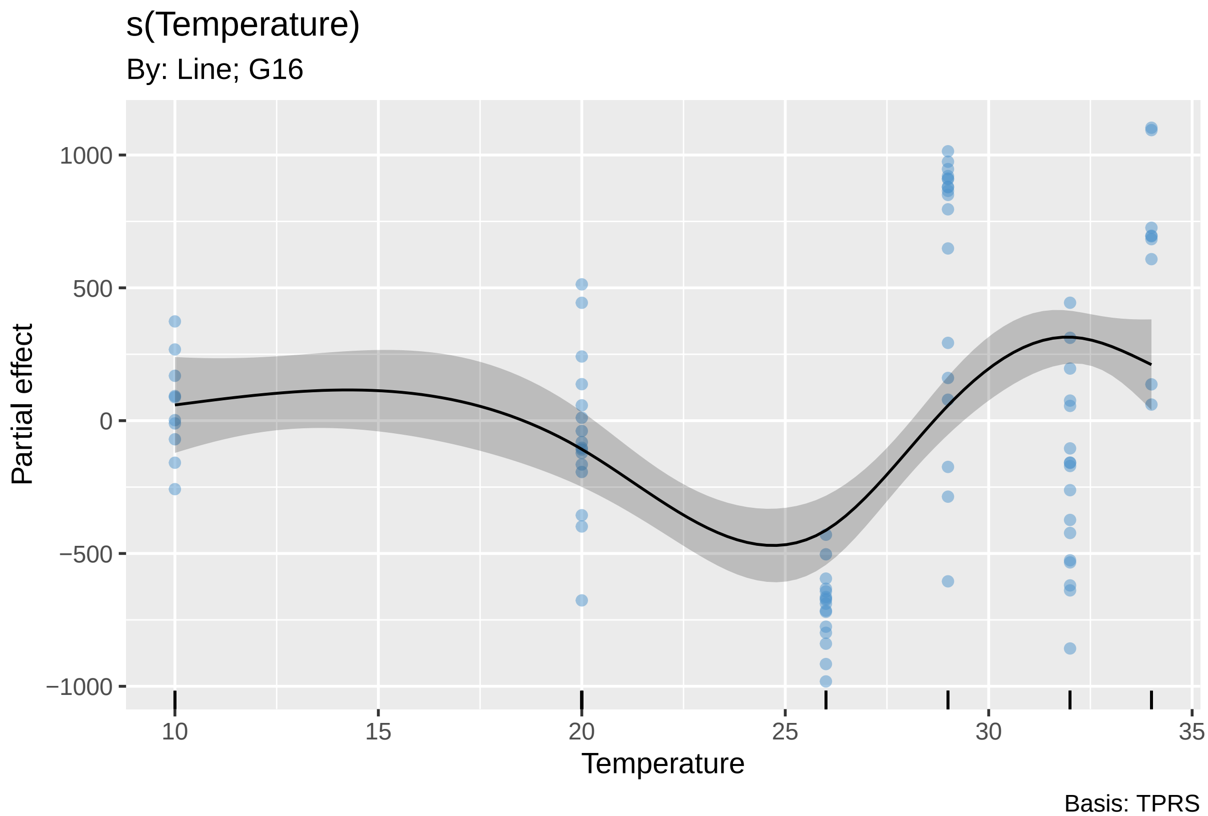

s(Temperature)
By: Line; G102
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS


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

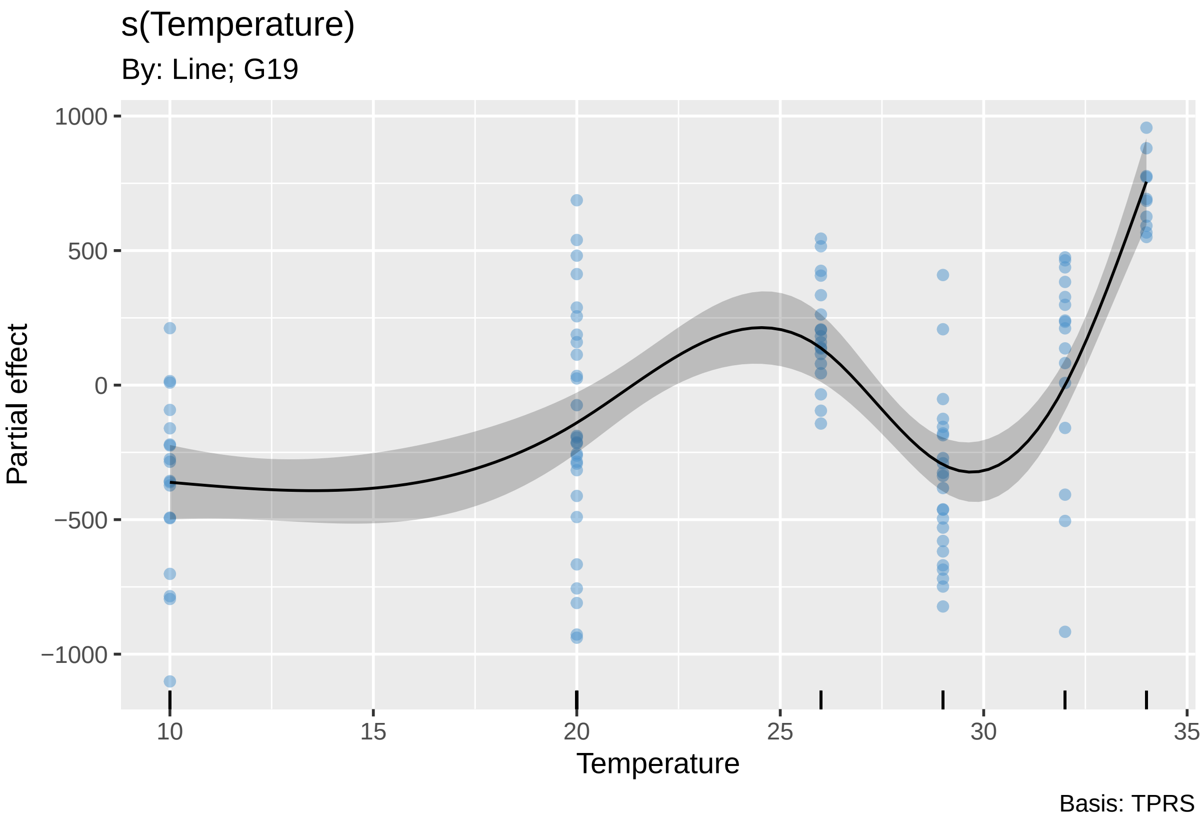


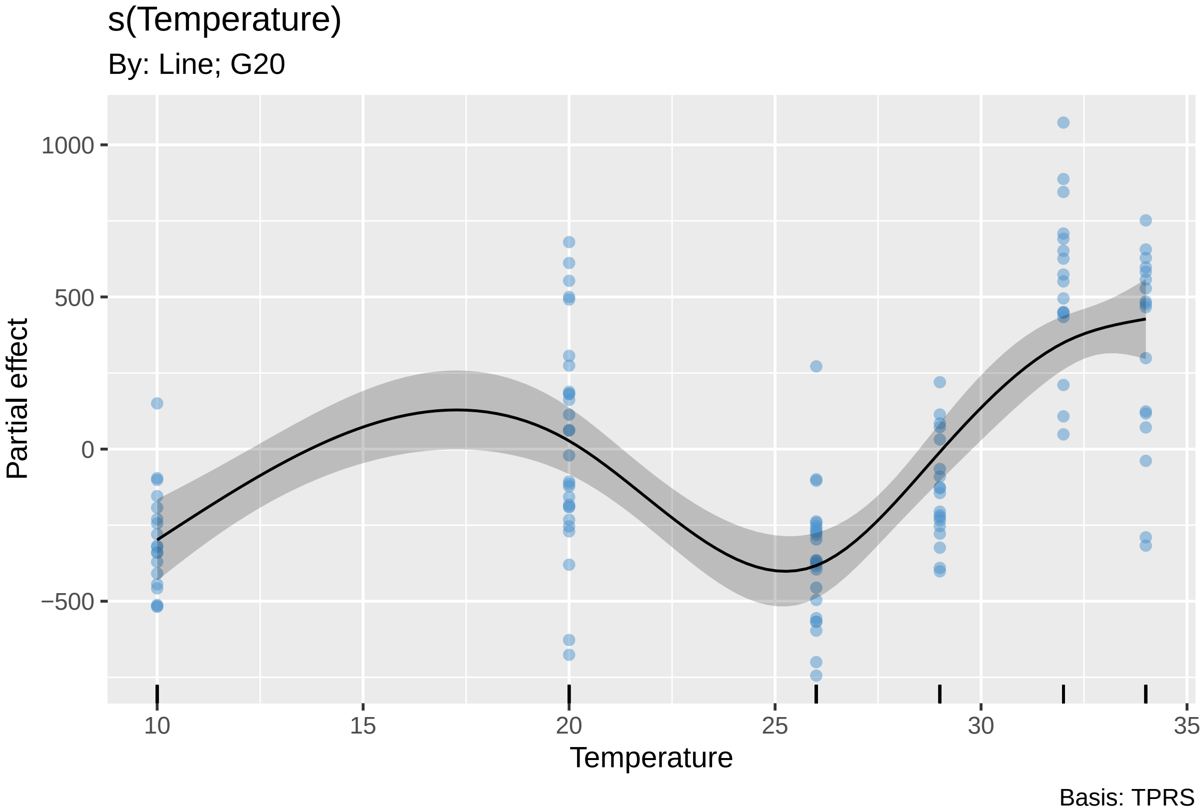
s(Temperature)
By: Line; G110
Partial effect
1000
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

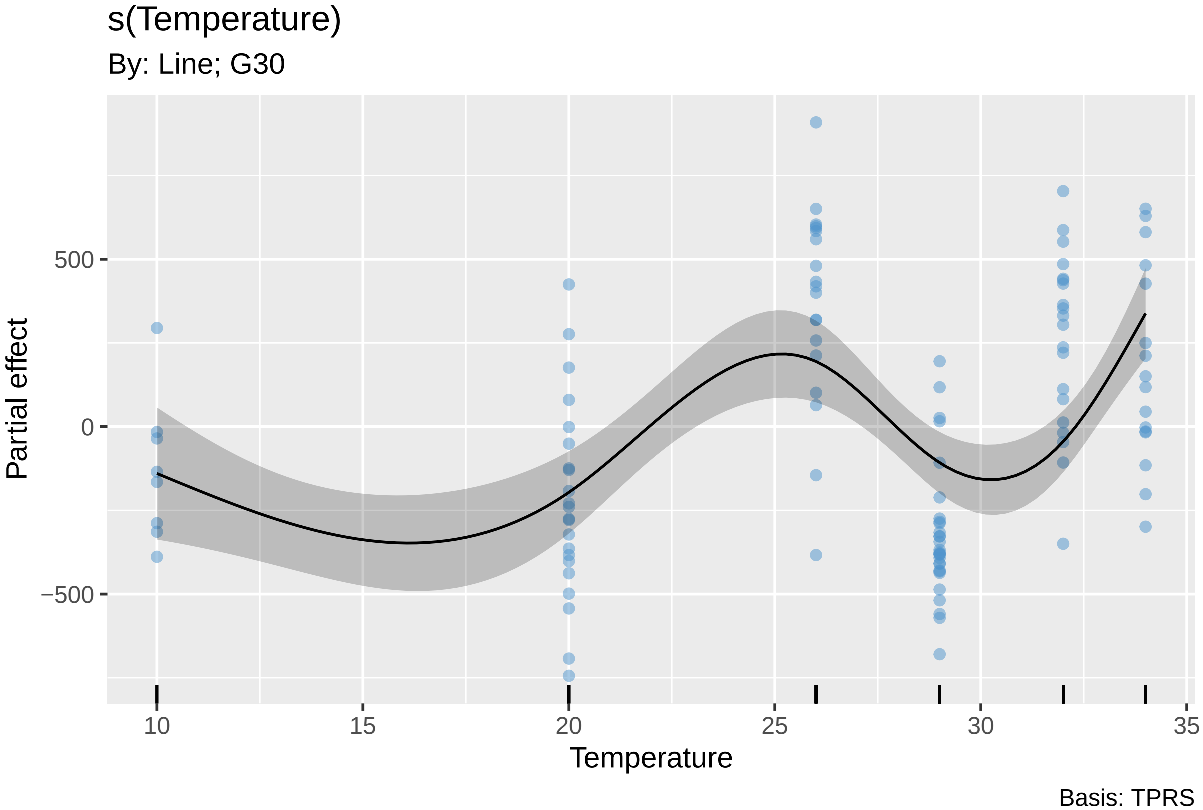

s(Temperature)
By: Line; G120
Partial effect
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

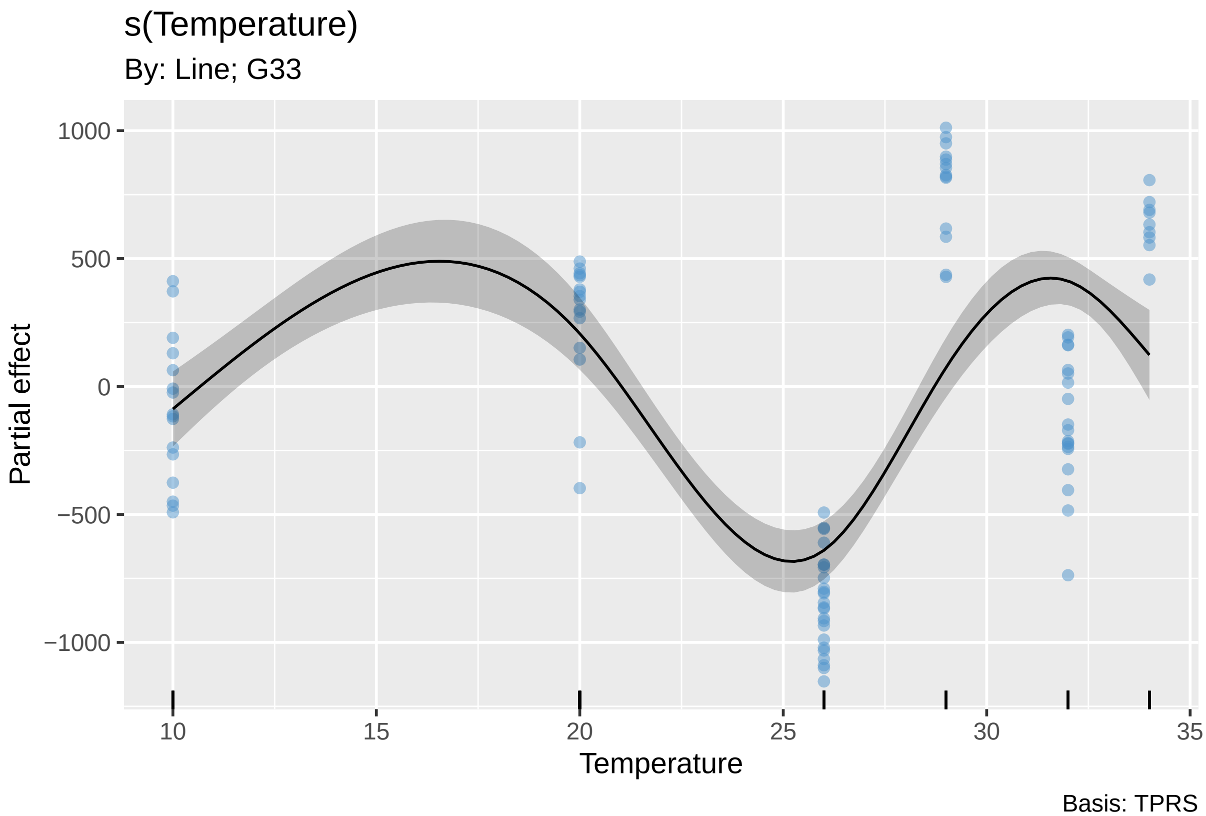


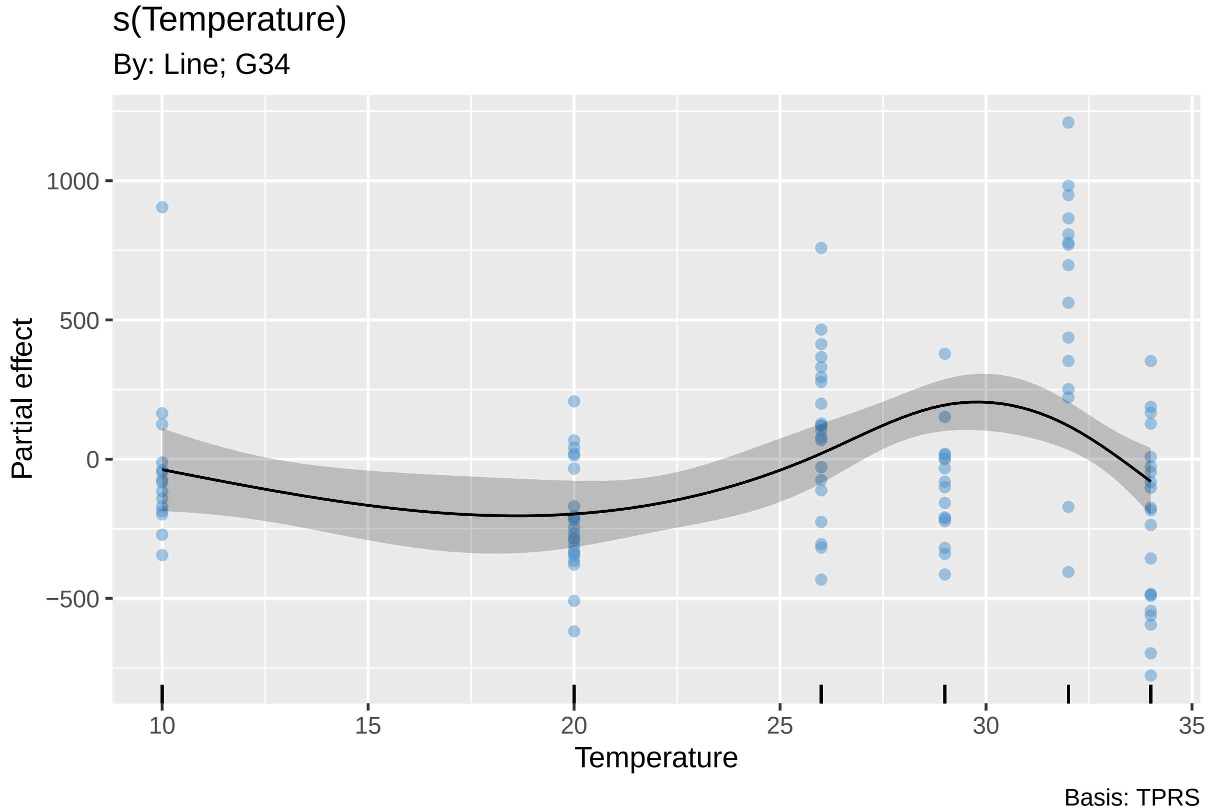
s(Temperature)
By: Line; G16
Partial effect
1000
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS


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

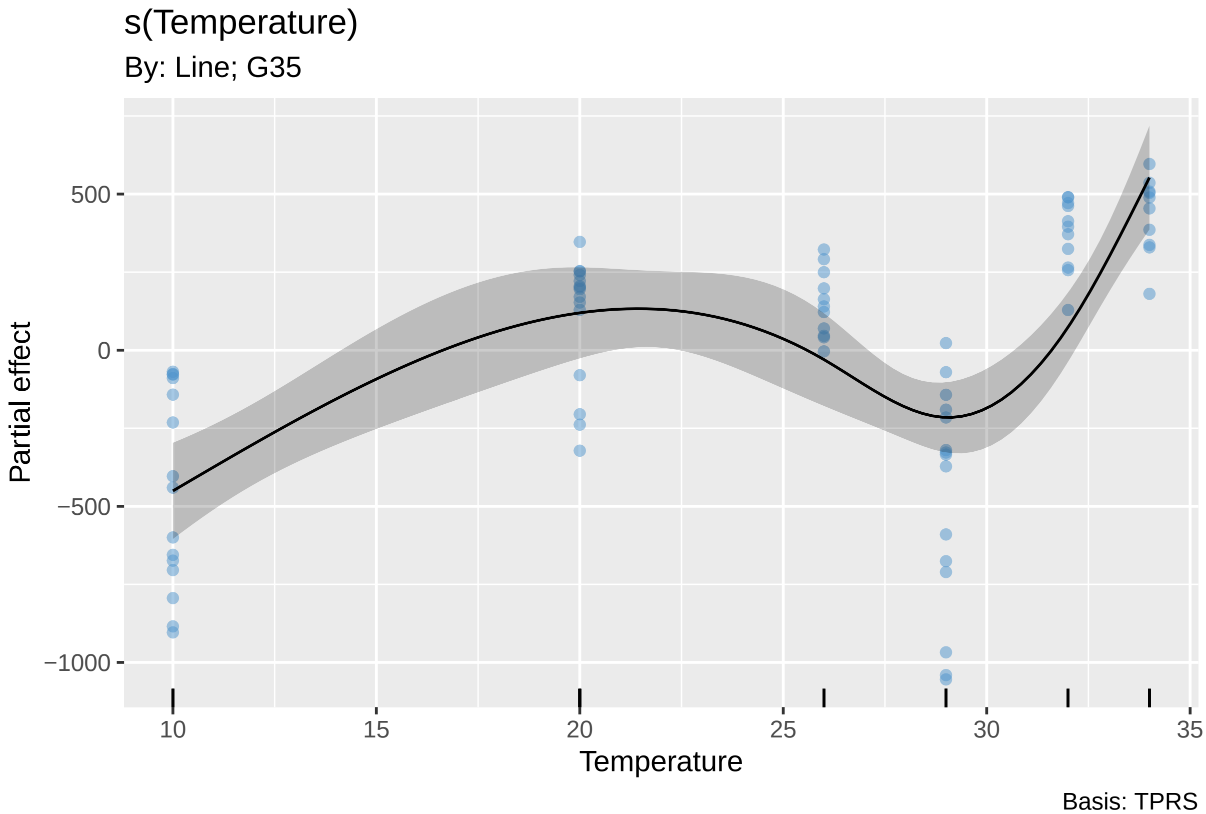

s(Temperature)
By: Line; G20
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS


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

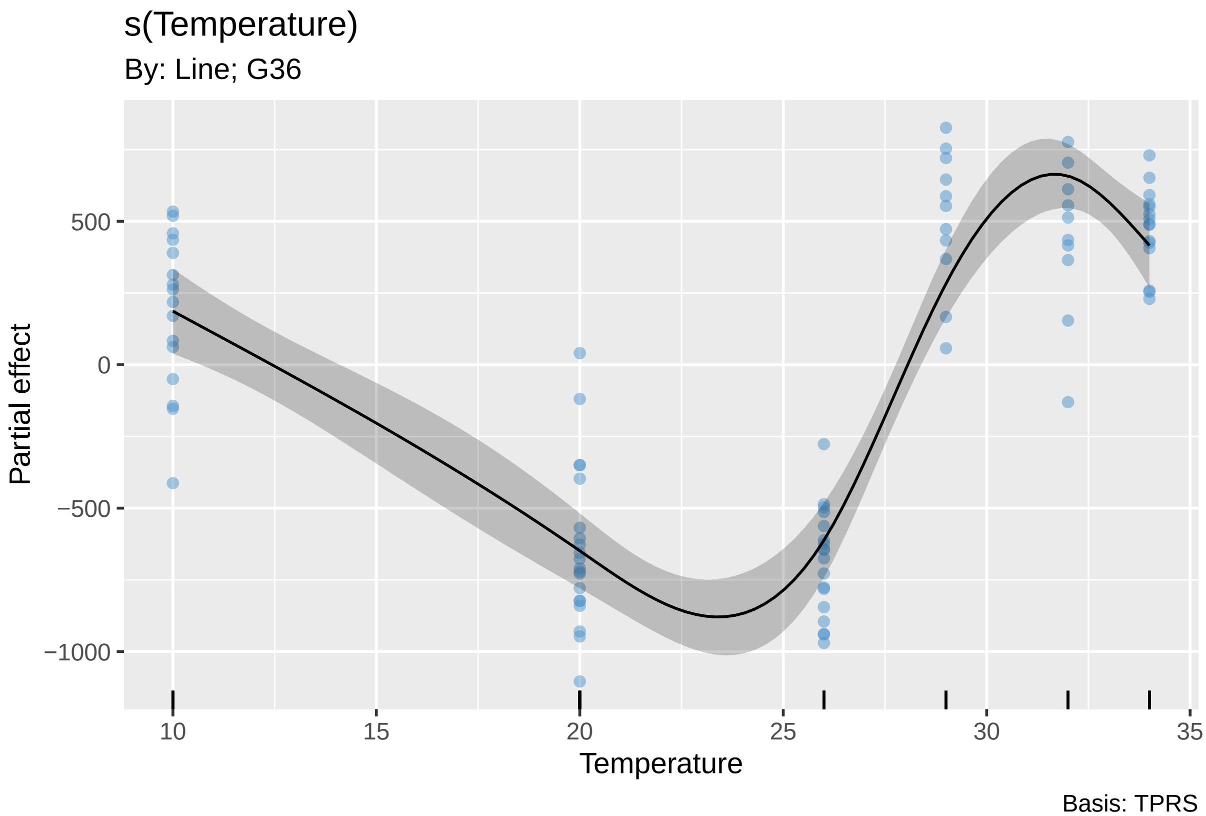

s(Temperature)
By: Line; G33
1000
500
0
−500
−1000
Partial effect
10
15
20
25
30
35
Temperature
Basis: TPRS

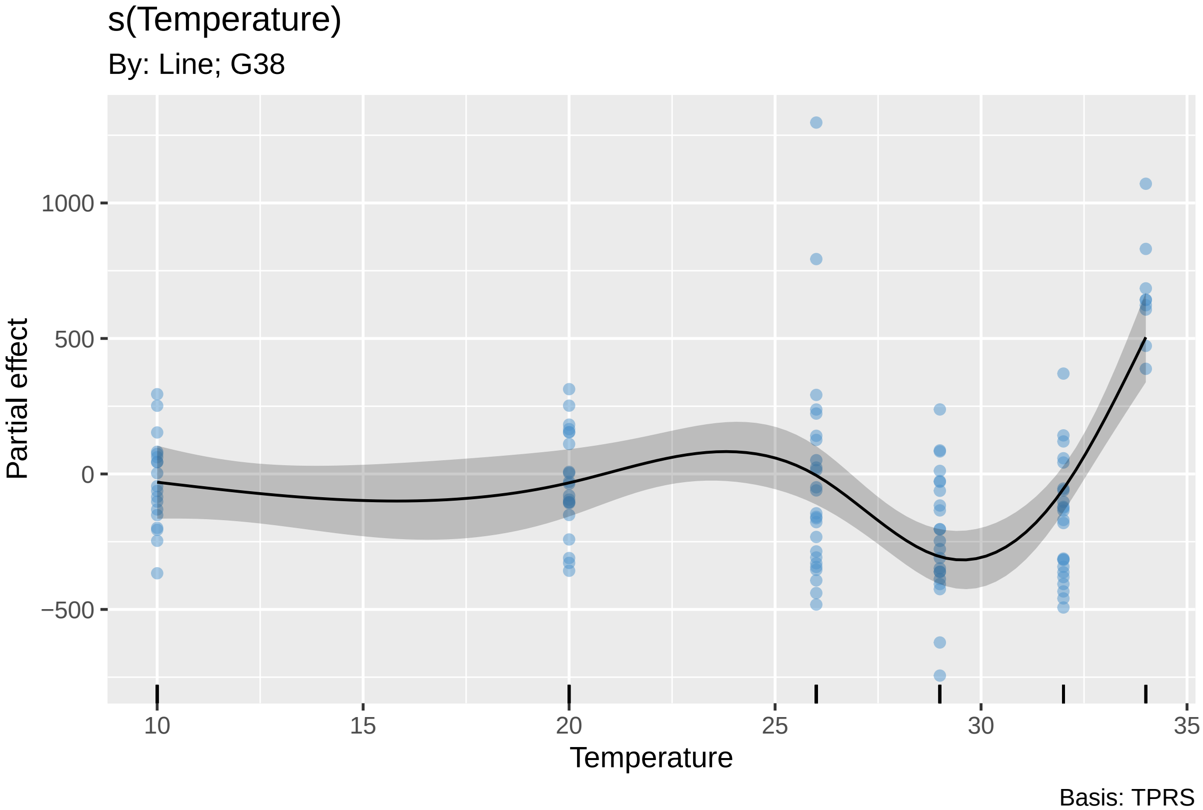


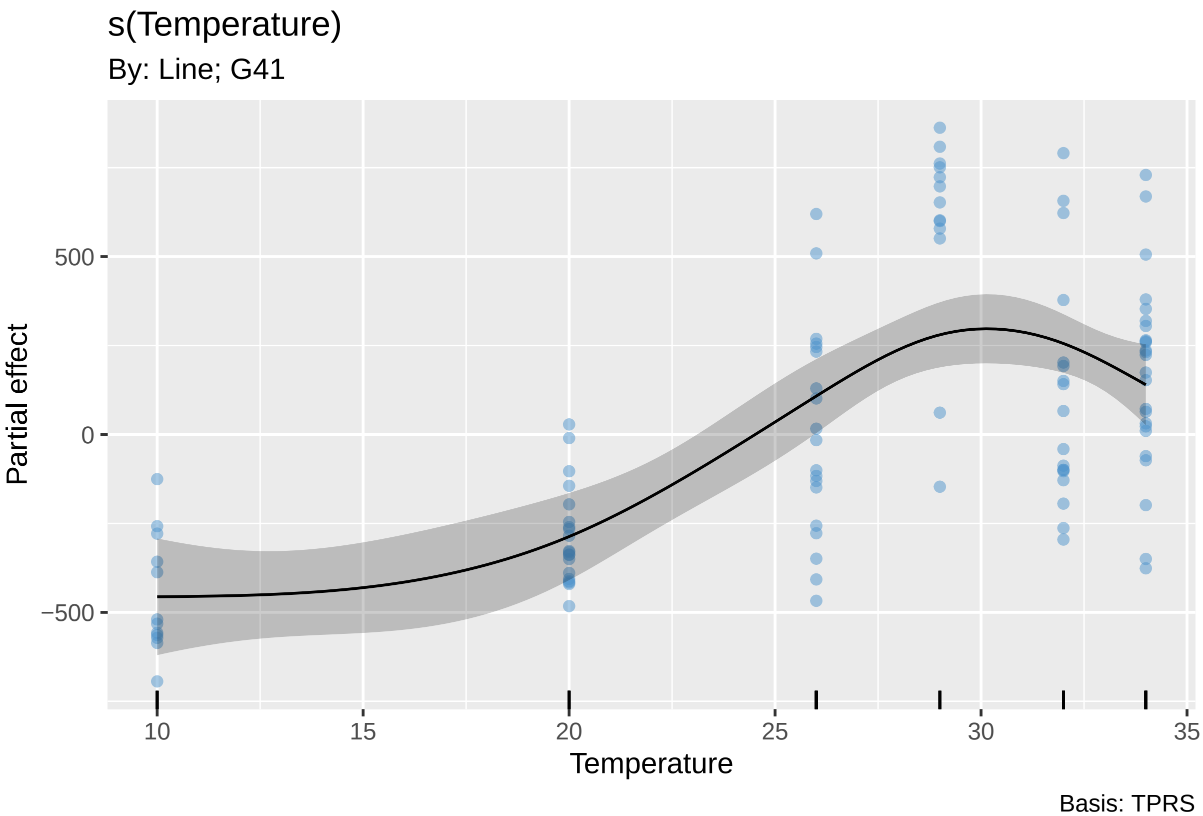
s(Temperature)
By: Line; G34
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

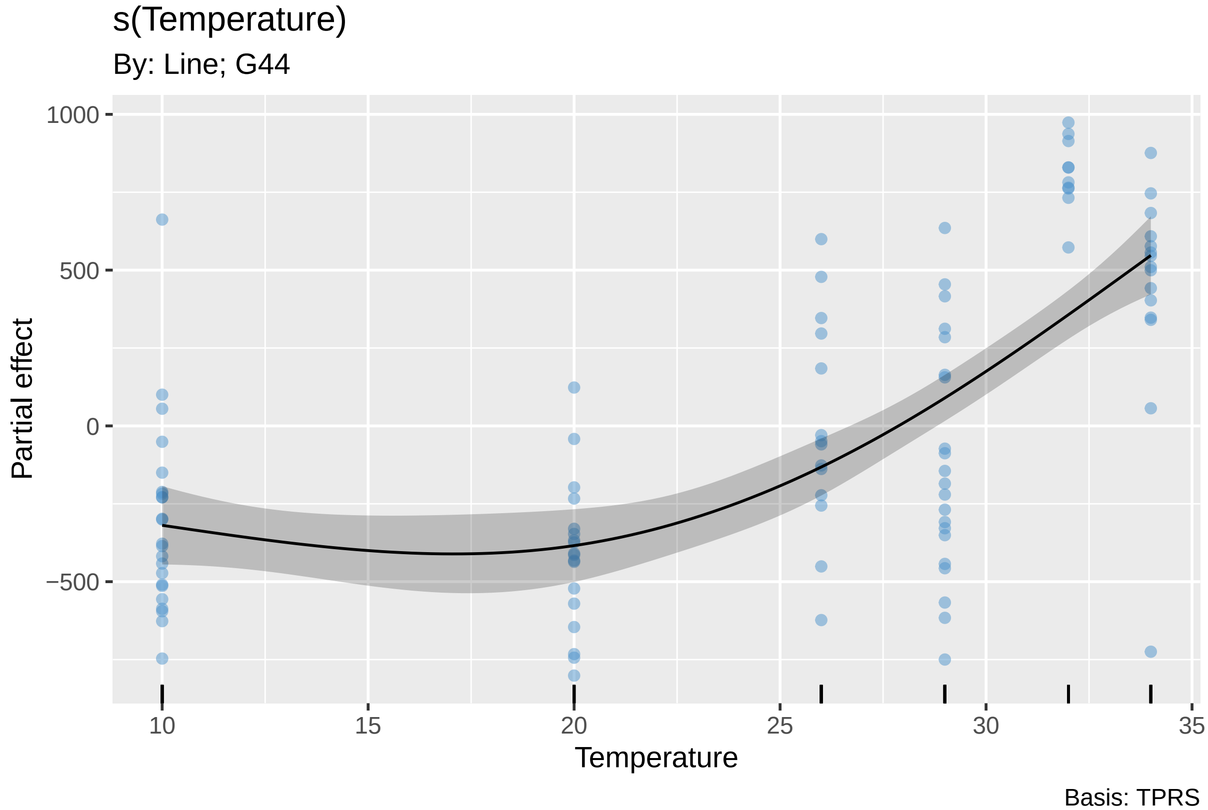


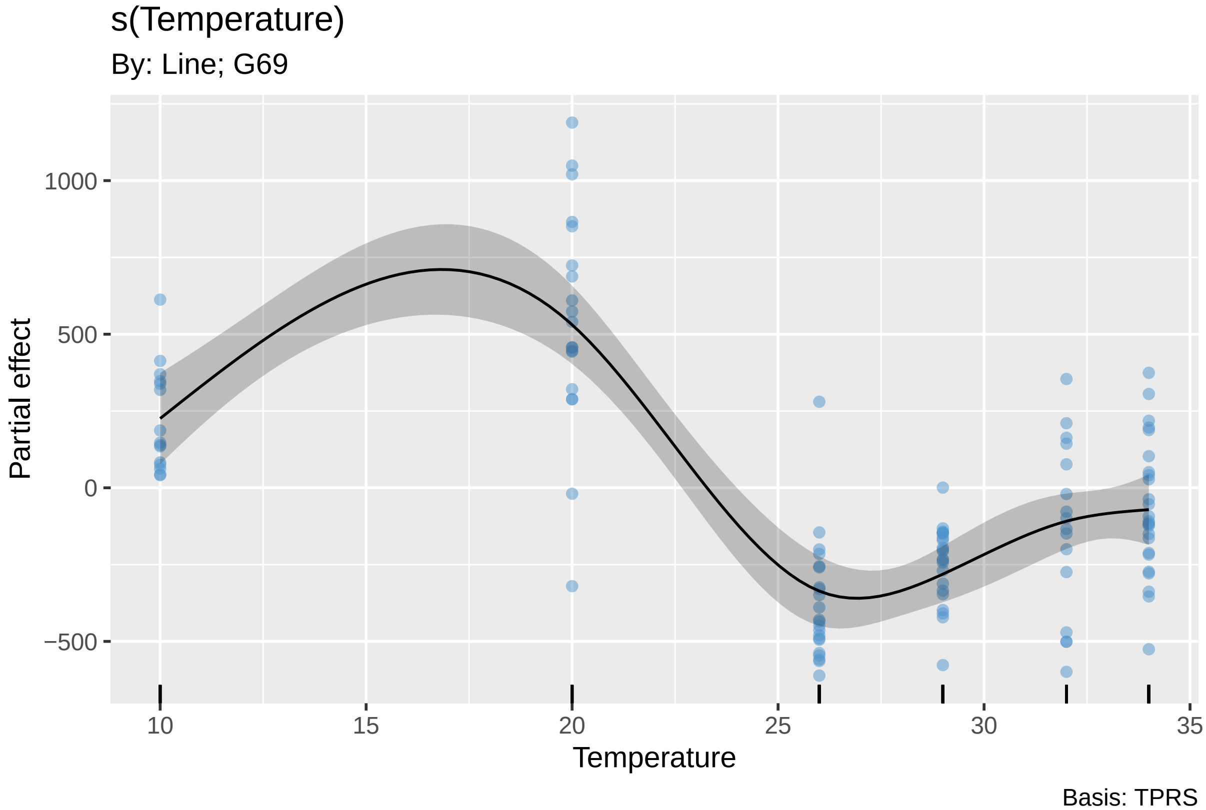
s(Temperature)
By: Line; G35
Partial effect
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

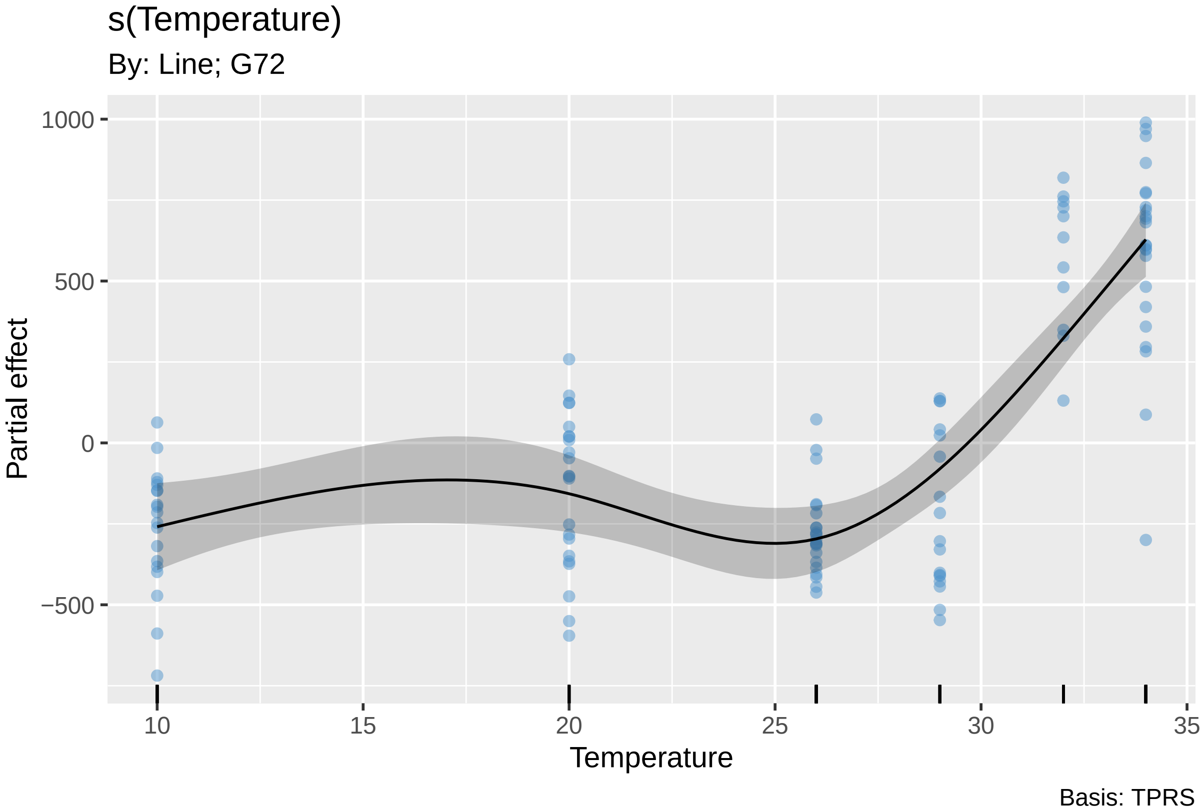

s(Temperature)
By: Line; G36
Partial effect
500
0
−500
−1000
10
15
20
25
30
35
Temperature
Basis: TPRS

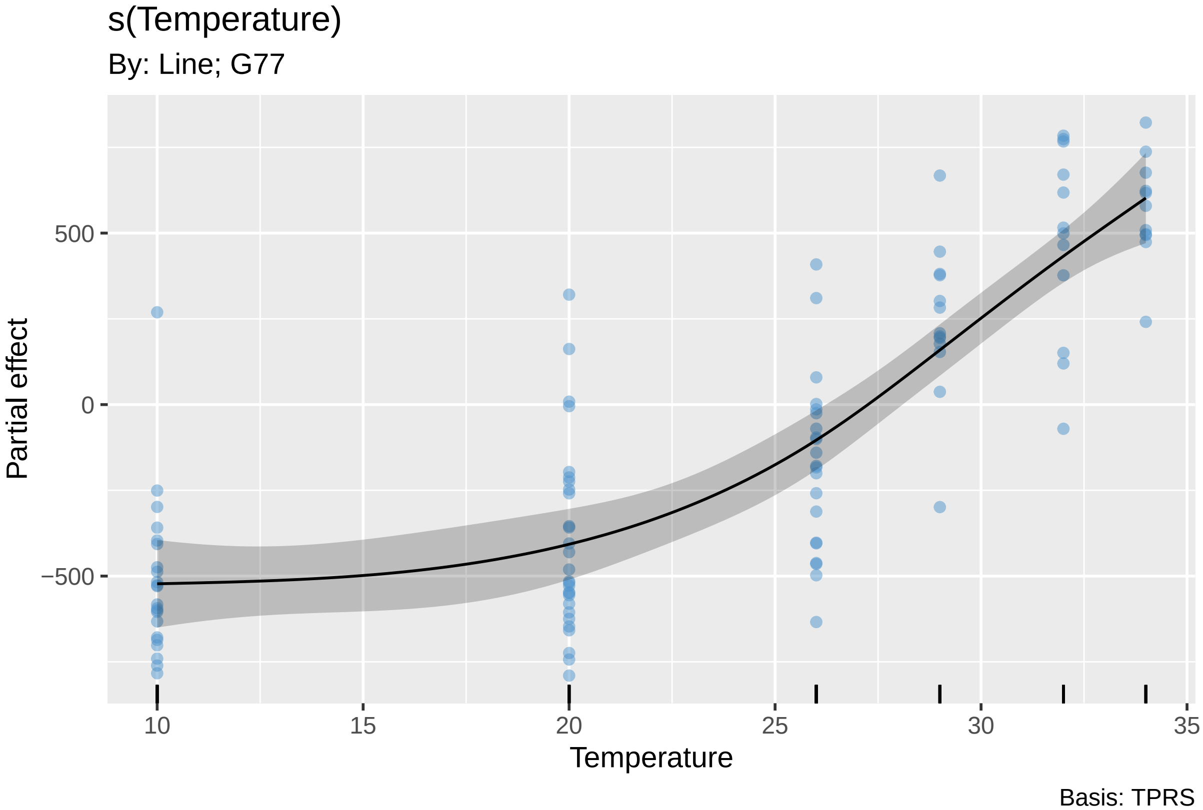

s(Temperature)
By: Line; G38
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS

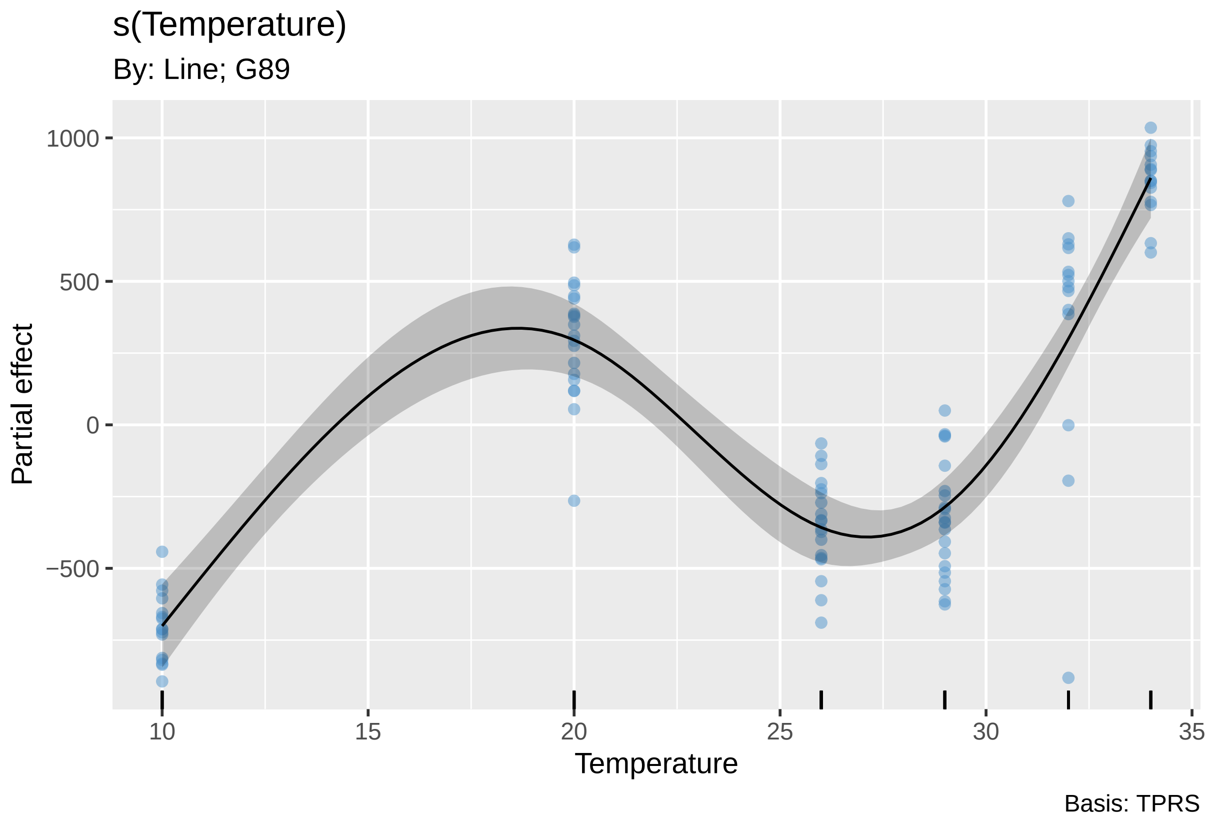

s(Temperature)
By: Line; G41
Partial effect
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS


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

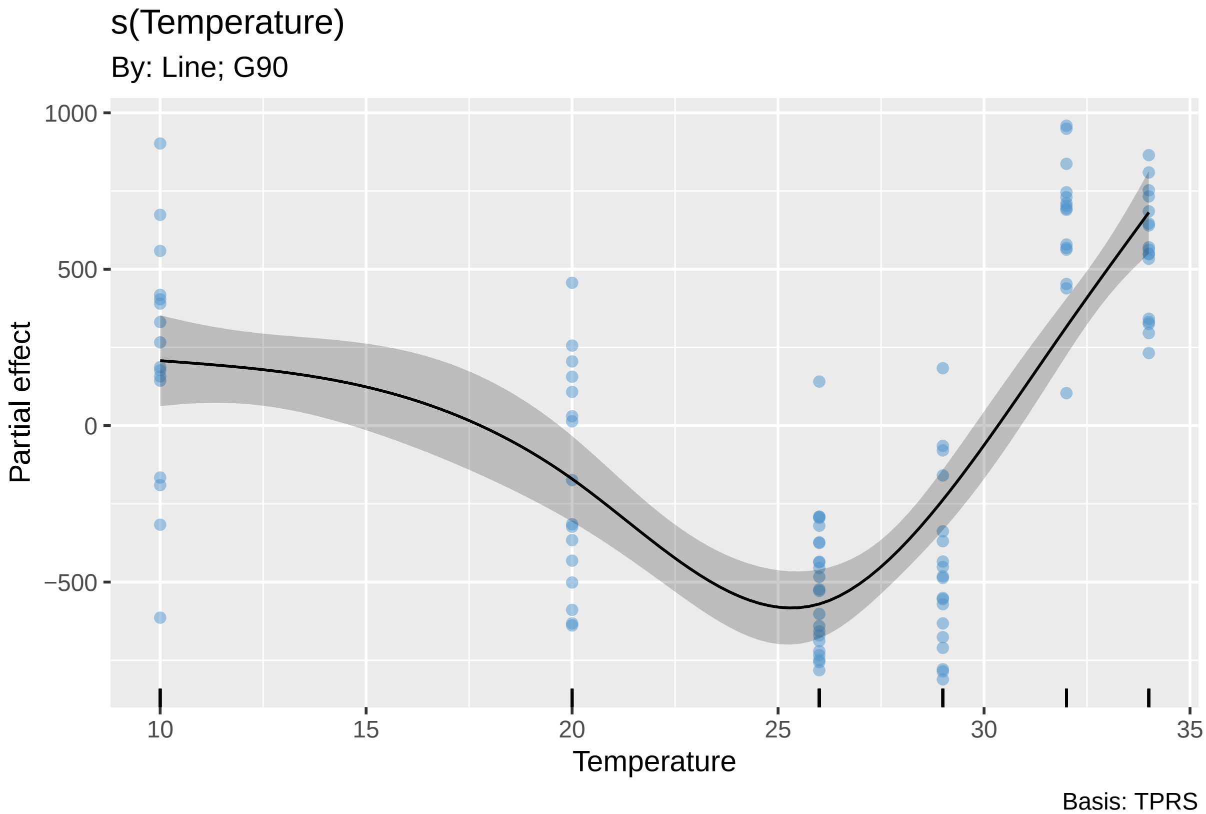

s(Temperature)
By: Line; G69
Partial effect
1000
500
0
−500
10
15
20
25
30
35
Temperature
Basis: TPRS


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
Partial effect
1000
500
0
−500
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 ($\Delta\text{AIC} = 941.6469014$).

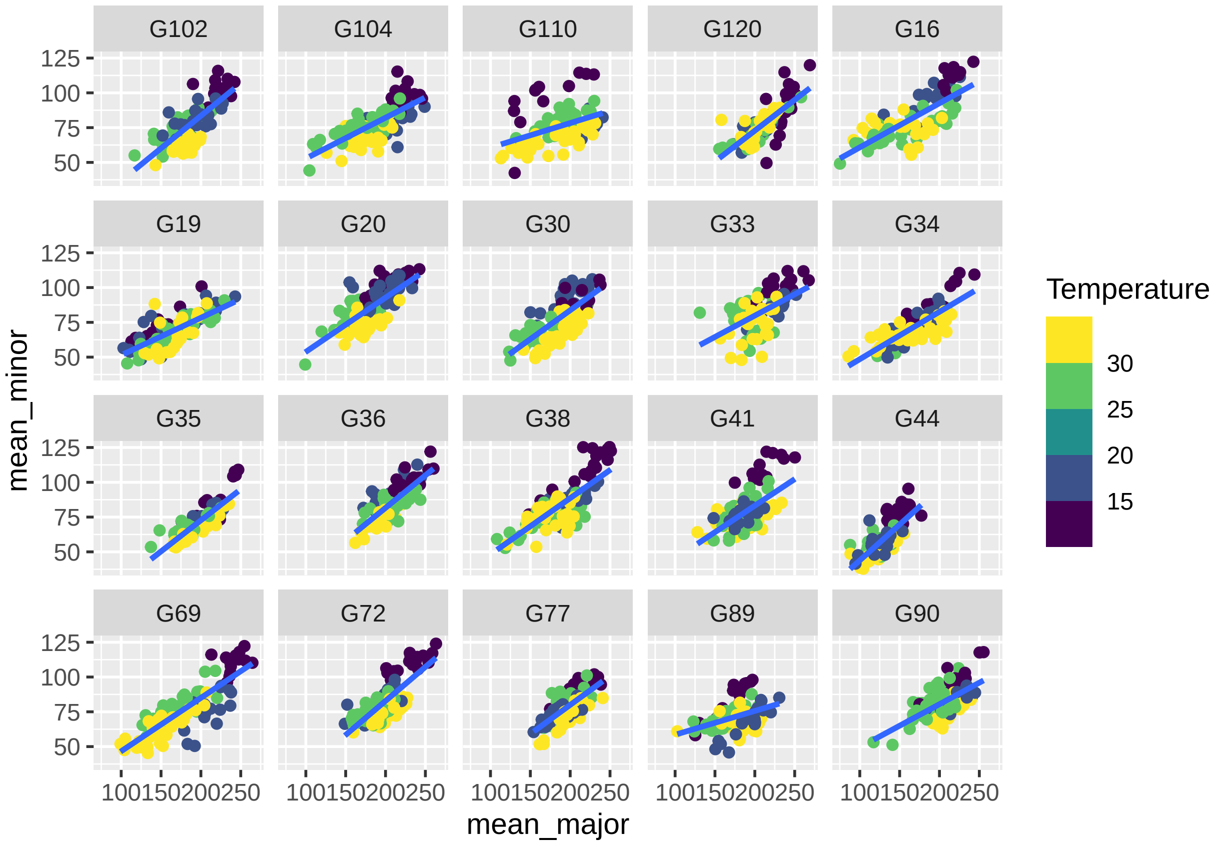
Examining the structure of plasticity in Paramecium

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 asssess 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'
```

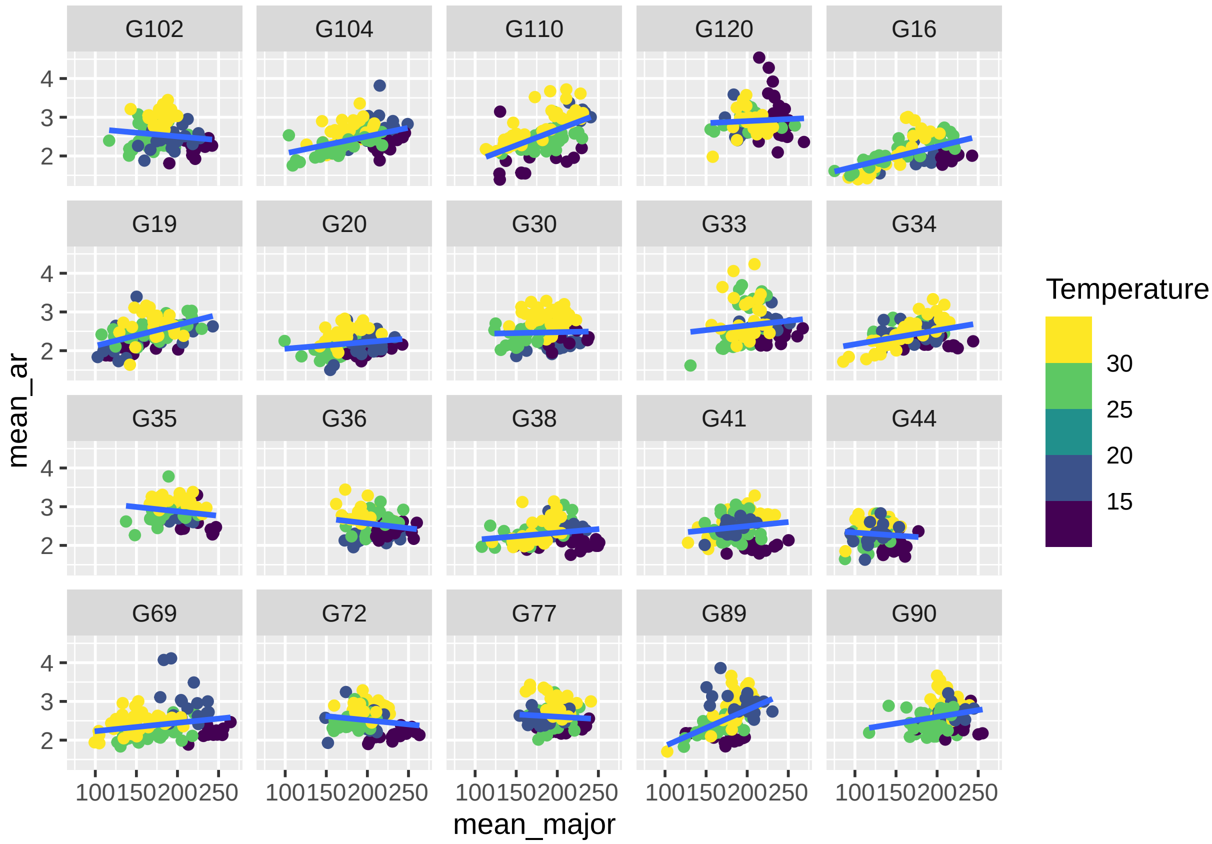


Across lines, it does appear to be the case that shorter cells are also less wide. Furthermore, looking at the relationships with temperature, many of the lines tend to have the longest and widest cells at low temperatures and the shortest, thinnest cells at high temperatures (e.g. G69). There also is 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'
```

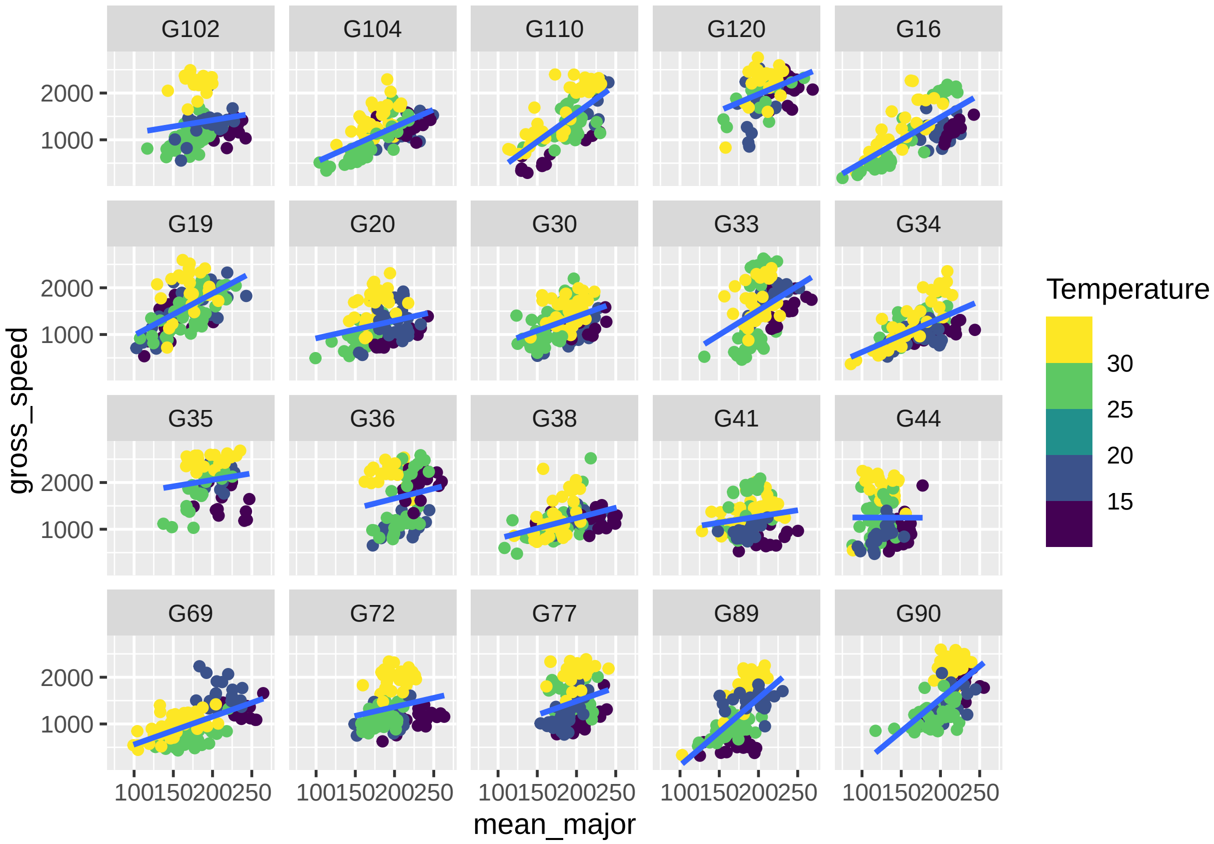


Here, we see far less evidence for a relationship with generally shallow slopes in the relationship between length and aspect ratio.

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'
```

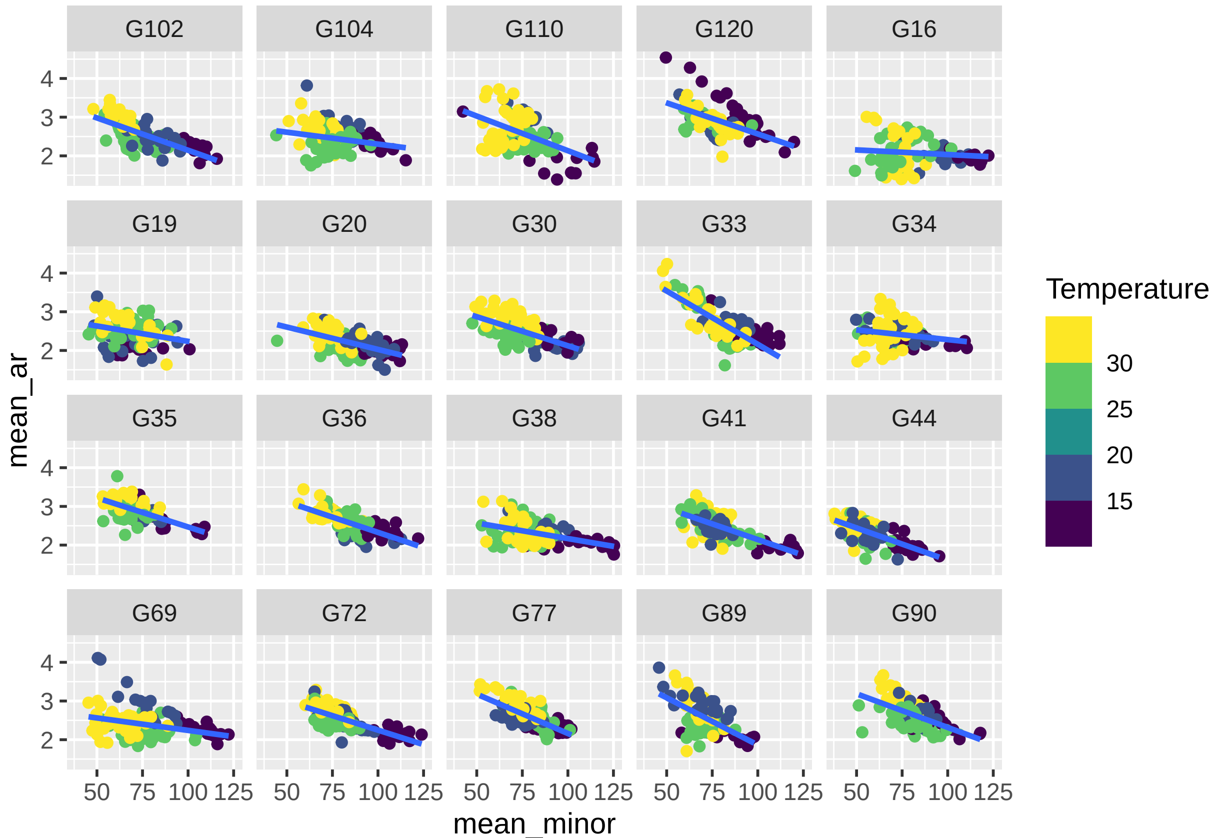


Here, we do see that some outcrossed lines seem to show a relationship between speed and length, but that isn't universal across all of the outcrossed lines.

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

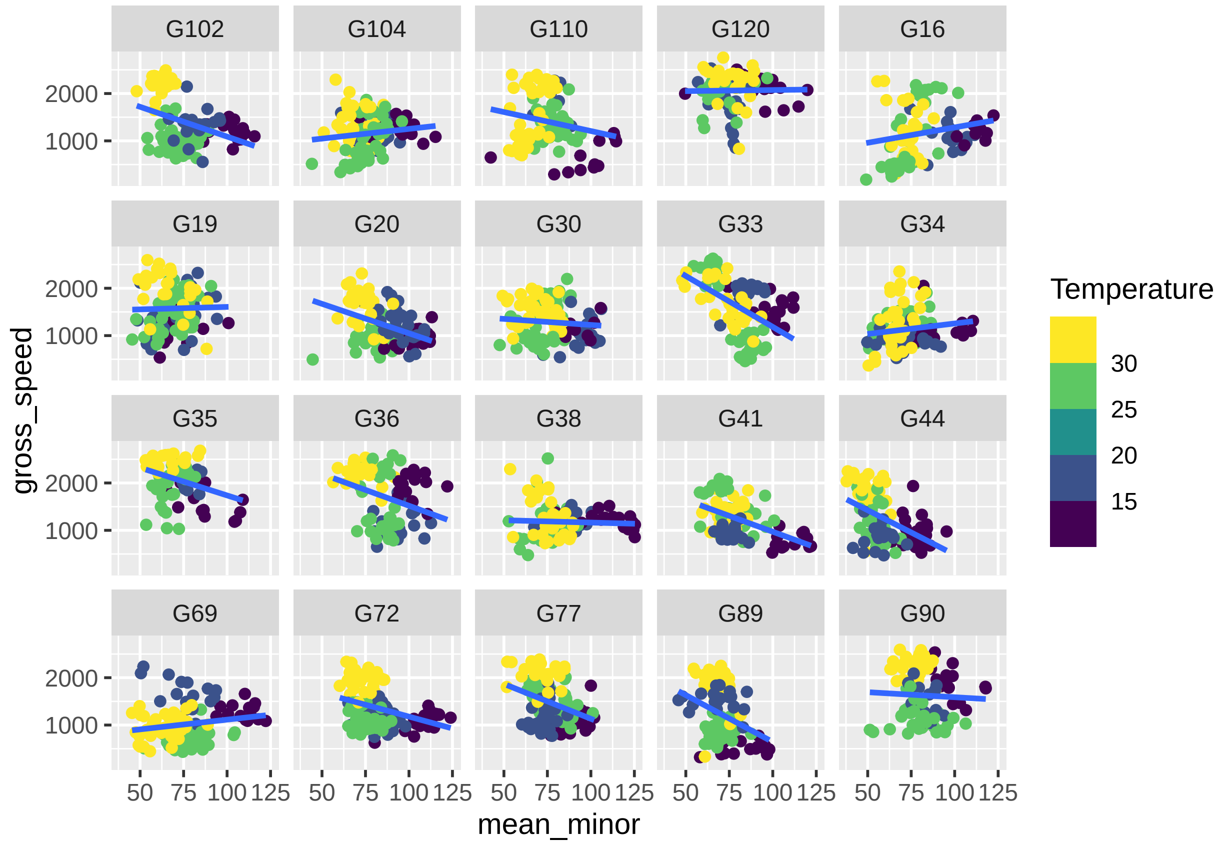
```
ggplot(data = morph_data, aes(x = mean_minor, y = mean_ar)) + geom_point(aes(color = Temperature)) + ge
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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




```r
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'
```

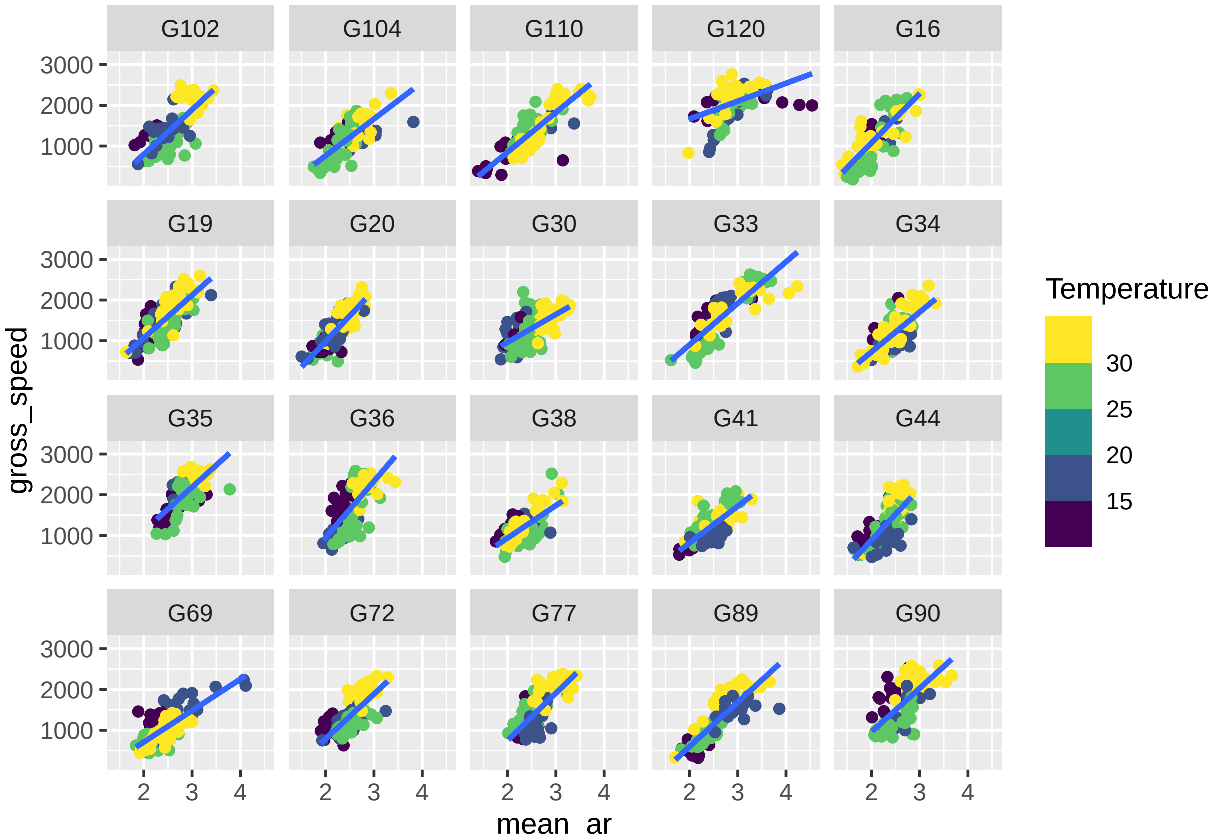


In the width and aspect ratio plots we see a weak negative relationship between width and aspect ratio as we might expect. This appears to translate to negative relationships between width and speed in some outcrossed lines but not others.

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)) + g
  facet_wrap(facets = "Line", nrow = 4, ncol = 5) +
  scale_color_viridis_b()
```

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



Here, we see fairly strong relationships between aspect ratio and speed as we had found previously across outcrossed lines. 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 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 variables.

Quantifying Plasticity

Now that we have looked at how the different outcrossed lines are responding to temperature in their morphology and movement, we can also use the GAMs to quantify the degree of plasticity in each of the phenotypes. The way that I will do this is by using the predicted responses from the GAMs and then extracting the range of the predicted phenotypes across the temperature range. I think that this provides a decent non-parametric measurement of plasticity in the phenotypes.

```r
### get predictions for length

newdata_length <- data.frame(Temperature = rep(seq(10, 34, by = 0.1),20),
                             Line = rep(unique(morph_data$Line), each = 241))

length_predictions <- predict.gam(gam_length, newdata = newdata_length)

newdata_length <- cbind(newdata_length, length_predictions)

length_predictions <- newdata_length %>% group_by(Line) %>%
  summarise(mean_length = mean(length_predictions),
            min_length = min(length_predictions),
```

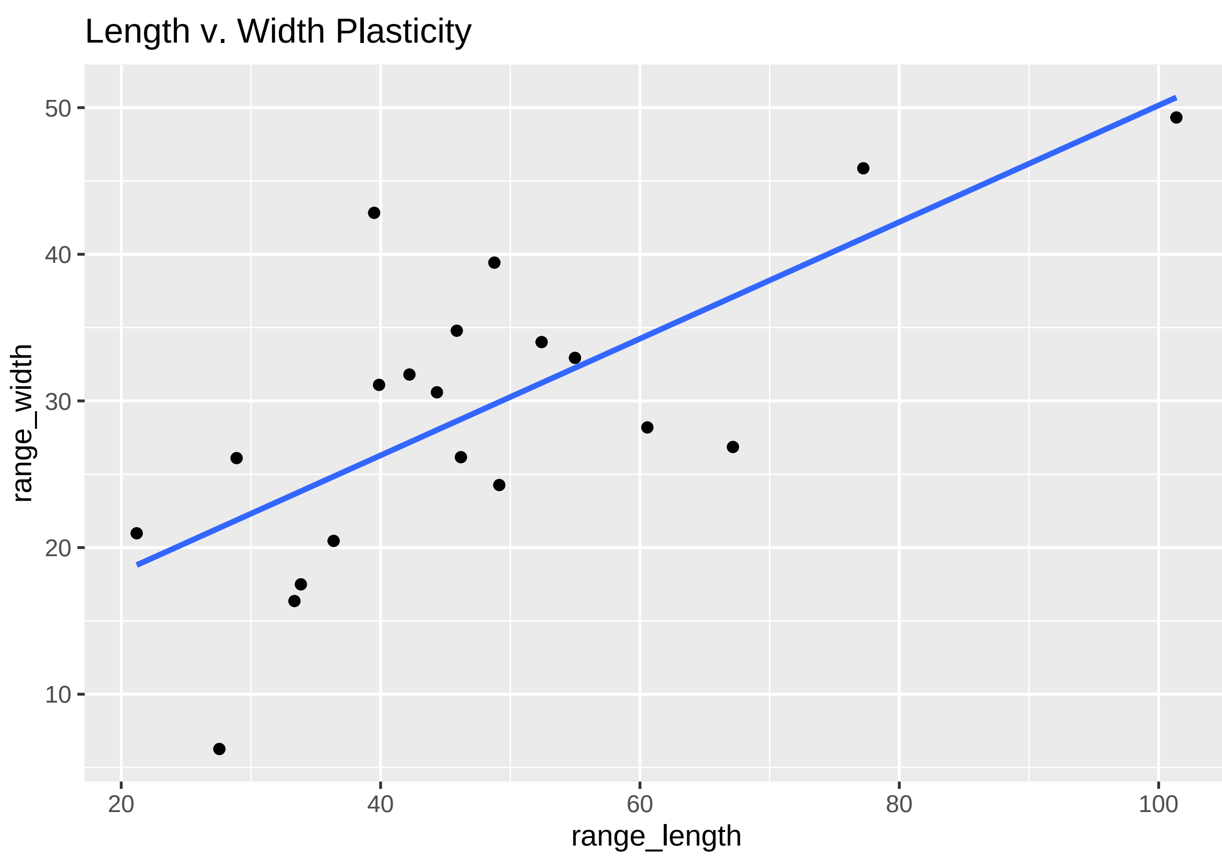
```r
            max_length = max(length_predictions),
            range_length = max_length-min_length,
            stand_range_length = range_length/max_length)

### predictions for width

newdata_width <- data.frame(Temperature = rep(seq(10, 34, by = 0.1),20),
                            Line = rep(unique(morph_data$Line), each = 241))

width_predictions <- predict.gam(gam_width, newdata = newdata_width)

newdata_width <- cbind(newdata_width, width_predictions)

width_predictions <- newdata_width %>% group_by(Line) %>%
  summarise(mean_width = mean(width_predictions),
            min_width = min(width_predictions),
            max_width = max(width_predictions),
            range_width = max_width-min_width,
            stand_range_width = range_width/max_width)

### predictions for aspect ratio

newdata_ar <- data.frame(Temperature = rep(seq(10, 34, by = 0.1),20),
                         Line = rep(unique(morph_data$Line), each = 241))

ar_predictions <- predict.gam(gam_ar, newdata = newdata_ar)

newdata_ar <- cbind(newdata_ar, ar_predictions)

ar_predictions <- newdata_ar %>% group_by(Line) %>%
  summarise(mean_ar = mean(ar_predictions),
            min_ar = min(ar_predictions),
            max_ar = max(ar_predictions),
            range_ar = max_ar-min_ar,
            stand_range_ar = range_ar/max_ar)

### predictions for speed

newdata_speed <- data.frame(Temperature = rep(seq(10, 34, by = 0.1),20),
                         Line = rep(unique(morph_data$Line), each = 241))

speed_predictions <- predict.gam(gam_speed, newdata = newdata_speed)

newdata_speed <- cbind(newdata_speed, speed_predictions)

speed_predictions <- newdata_speed %>% group_by(Line) %>%
  summarise(mean_speed = mean(speed_predictions),
            min_speed = min(speed_predictions),
            max_speed = max(speed_predictions),
            range_speed = max_speed-min_speed,
            stand_range_speed = range_speed/max_speed)

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

```r
plast_morph_data <- full_join(length_predictions, width_predictions, by = "Line")

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

plast_morph_data <- full_join(plast_morph_data, speed_predictions, 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?

```r
### 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'
```



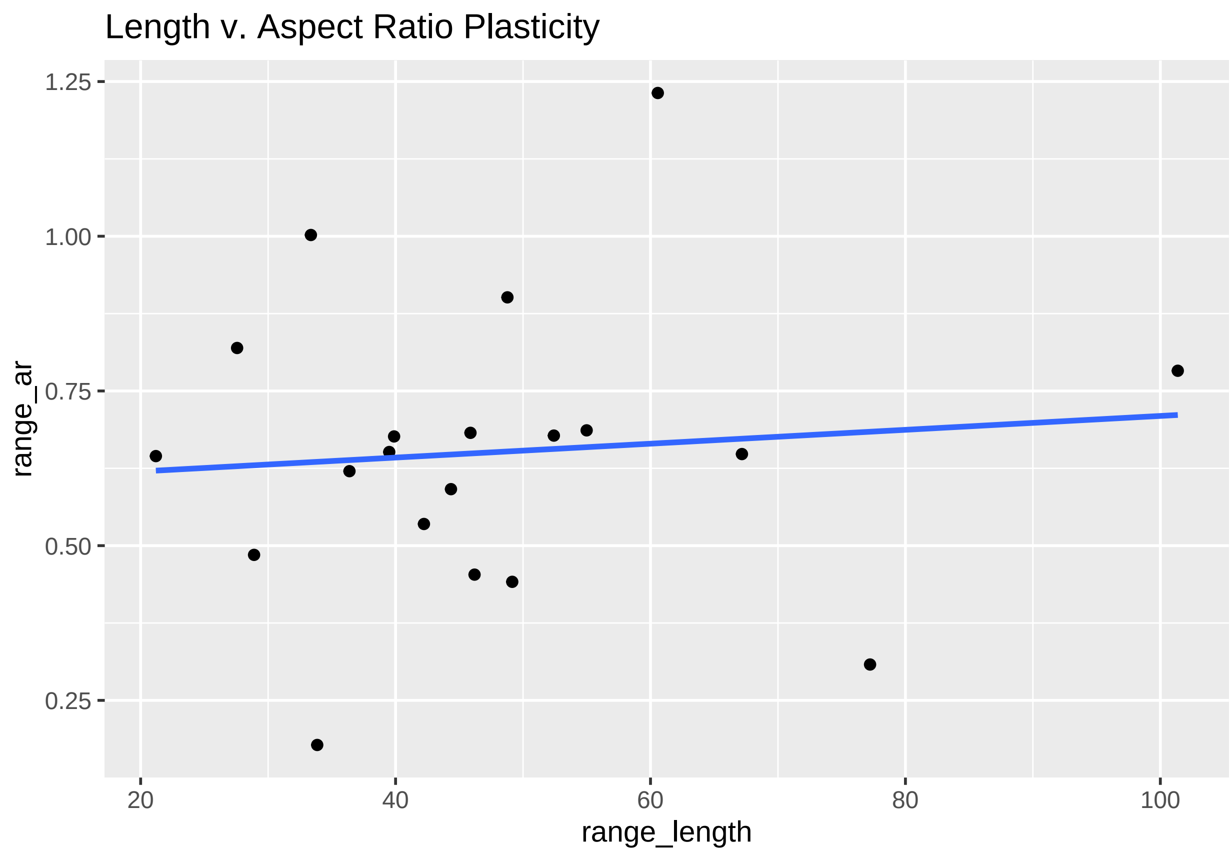

```r
summary(lm(range_width ~ range_length, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_width ~ range_length, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.079  -5.818   1.427   4.663  16.737
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.37626    4.78573   2.168 0.043794 *
## range_length  0.39778    0.09408   4.228 0.000506 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.607 on 18 degrees of freedom
## Multiple R-squared:  0.4983, Adjusted R-squared:  0.4704
## F-statistic: 17.88 on 1 and 18 DF,  p-value: 0.0005056
```

```
### 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'
```



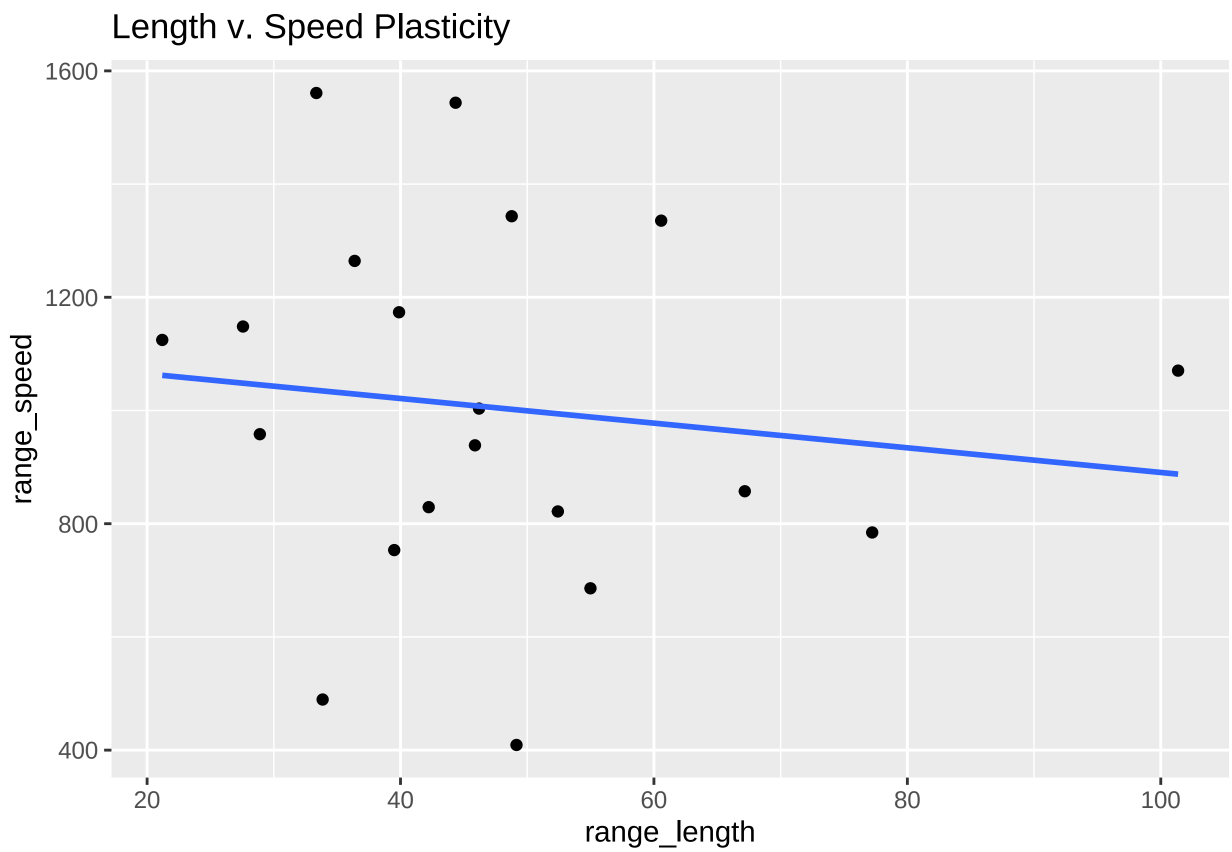

```
summary(lm(range_ar ~ range_length, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_ar ~ range_length, data = plast_morph_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.45755 -0.11856  0.01557  0.04359  0.56607
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.597503   0.150788   3.963 0.000913 ***
## range_length 0.001121   0.002964   0.378 0.709663
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2397 on 18 degrees of freedom
## Multiple R-squared:  0.007886,   Adjusted R-squared:  -0.04723
## F-statistic: 0.1431 on 1 and 18 DF,  p-value: 0.7097
```

```
### 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'
```



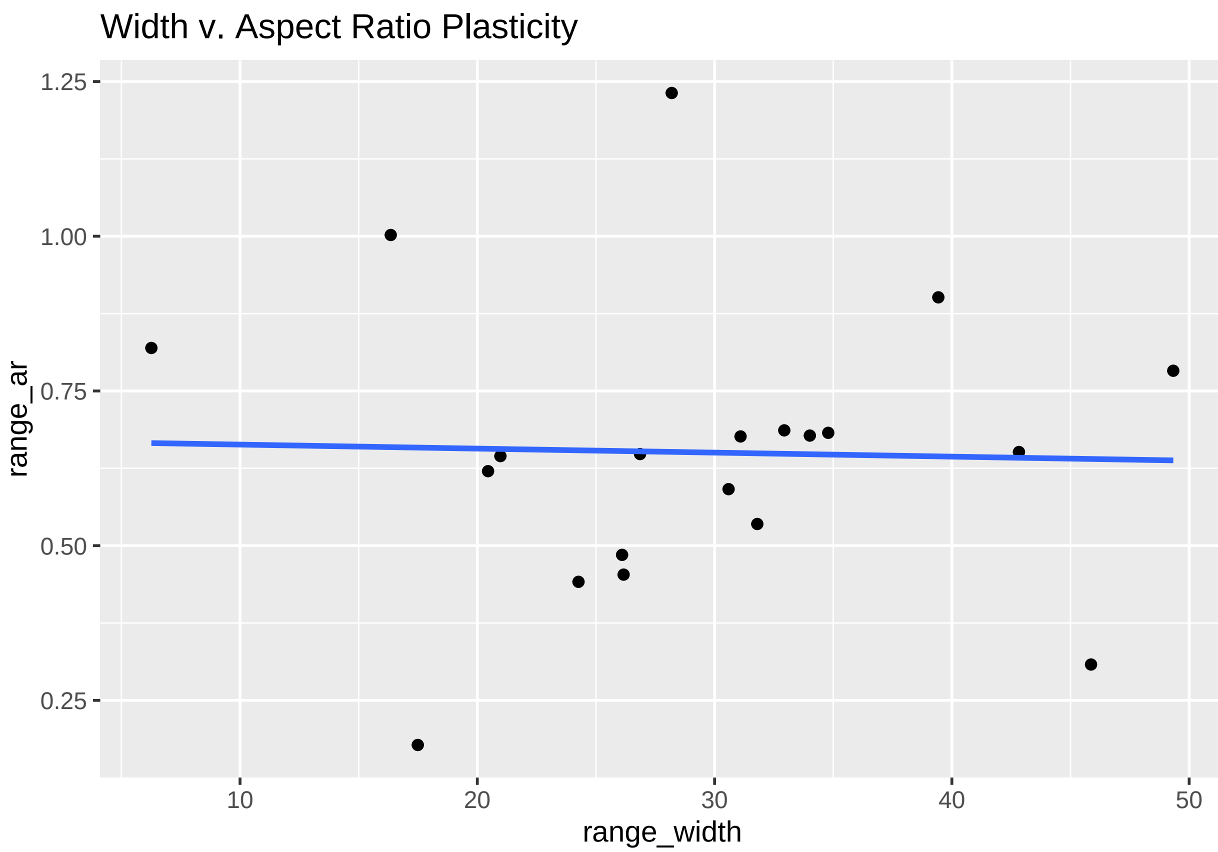

```
summary(lm(range_speed ~ range_length, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_speed ~ range_length, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -592.15 -176.14  -37.11  195.96  531.97
##
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1108.297    201.931   5.489 3.27e-05 ***
## range_length   -2.178      3.970  -0.549     0.59
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 321 on 18 degrees of freedom
## Multiple R-squared:  0.01644,    Adjusted R-squared:  -0.0382
## F-statistic: 0.3009 on 1 and 18 DF,  p-value: 0.59
```

```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'
```



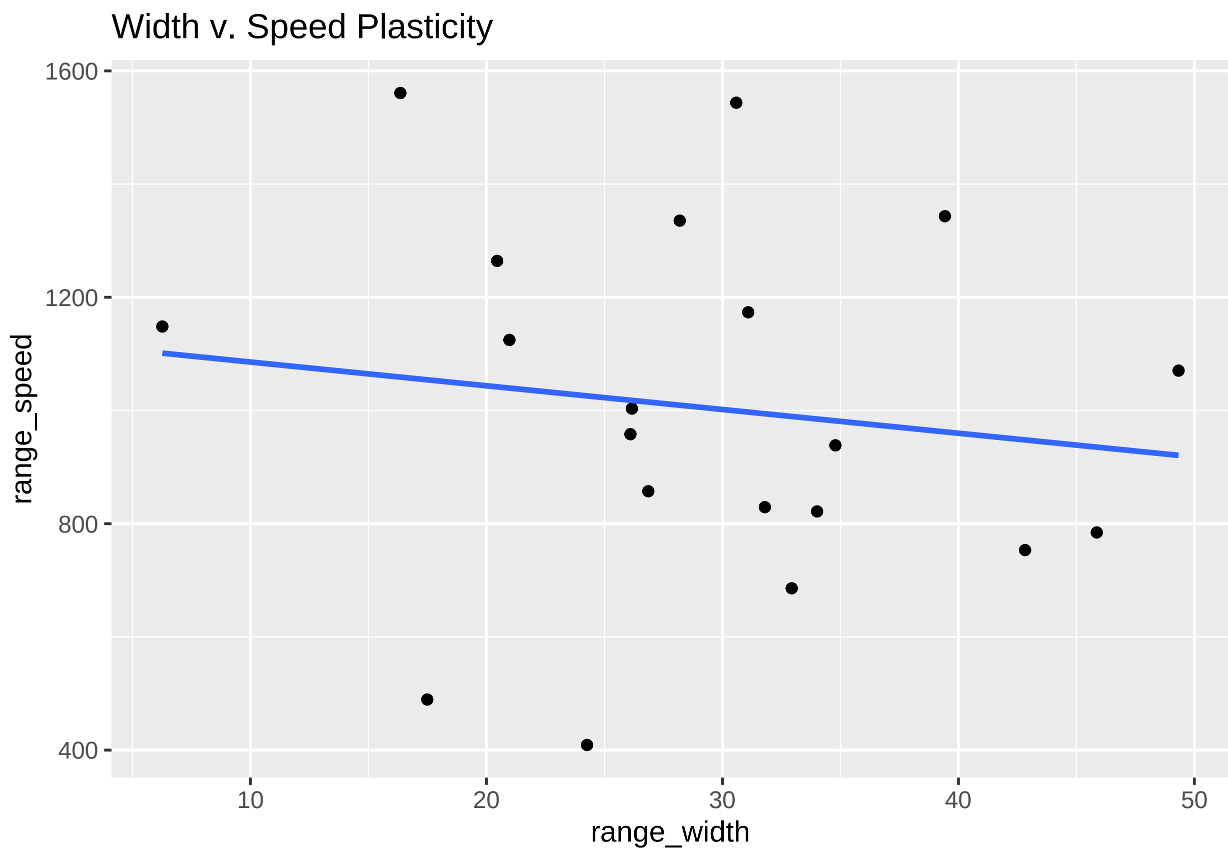

```r
summary(lm(range_ar ~ range_width, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_ar ~ range_width, data = plast_morph_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48055 -0.12756  0.00245  0.06460  0.57996
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6698048  0.1637062   4.092 0.000685 ***
## range_width -0.0006483  0.0052790  -0.123 0.903619
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2405 on 18 degrees of freedom
## Multiple R-squared:  0.0008372,  Adjusted R-squared:  -0.05467
## F-statistic: 0.01508 on 1 and 18 DF,  p-value: 0.9036
```

```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'
```



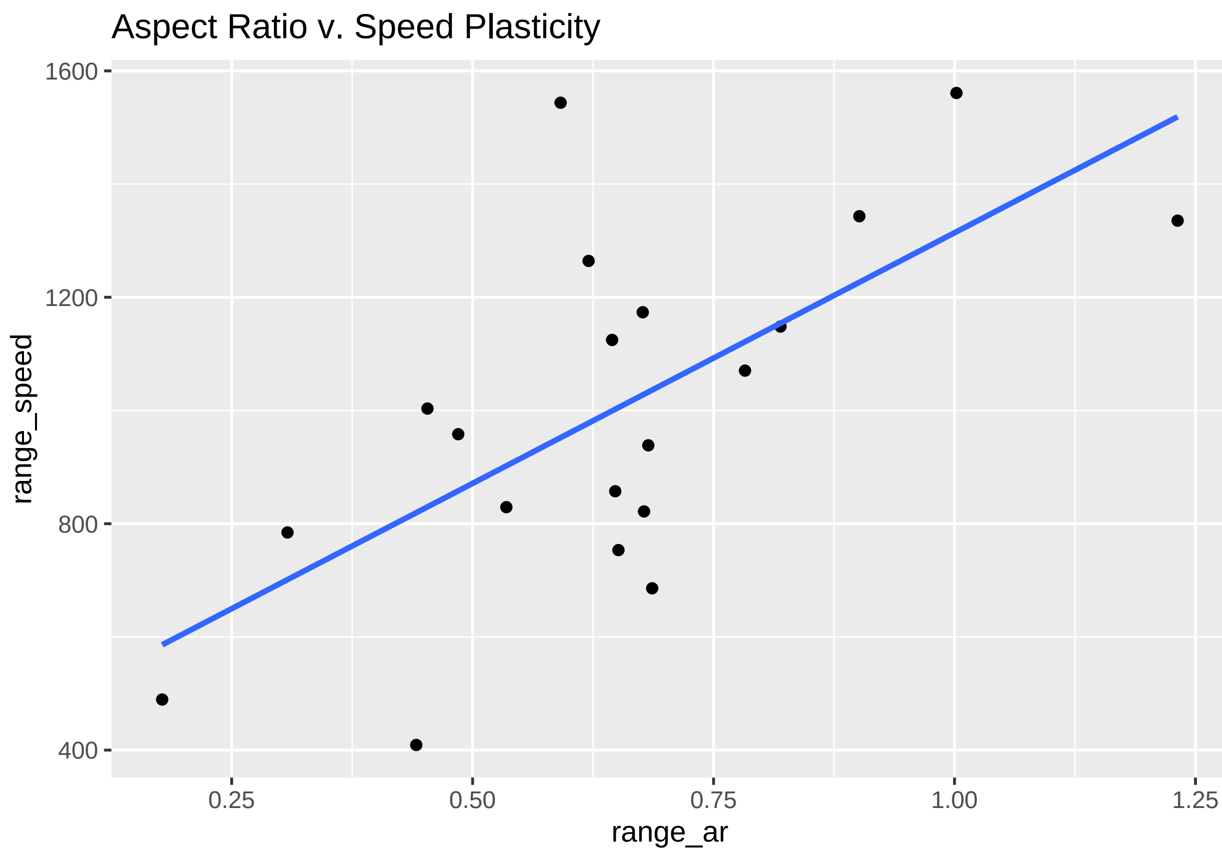

```r
summary(lm(range_speed ~ range_width, data = plast_morph_data))
```

```
##
## Call:
## lm(formula = range_speed ~ range_width, data = plast_morph_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -616.72 -163.71  -28.84  187.89  544.39
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1127.559    218.132   5.169 6.45e-05 ***
## range_width   -4.193      7.034  -0.596    0.559
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 320.5 on 18 degrees of freedom
## Multiple R-squared:  0.01936,    Adjusted R-squared:  -0.03512
## F-statistic: 0.3553 on 1 and 18 DF,  p-value: 0.5586
```

```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
## -410.35 -154.57  -28.35  130.48  591.62
##
```

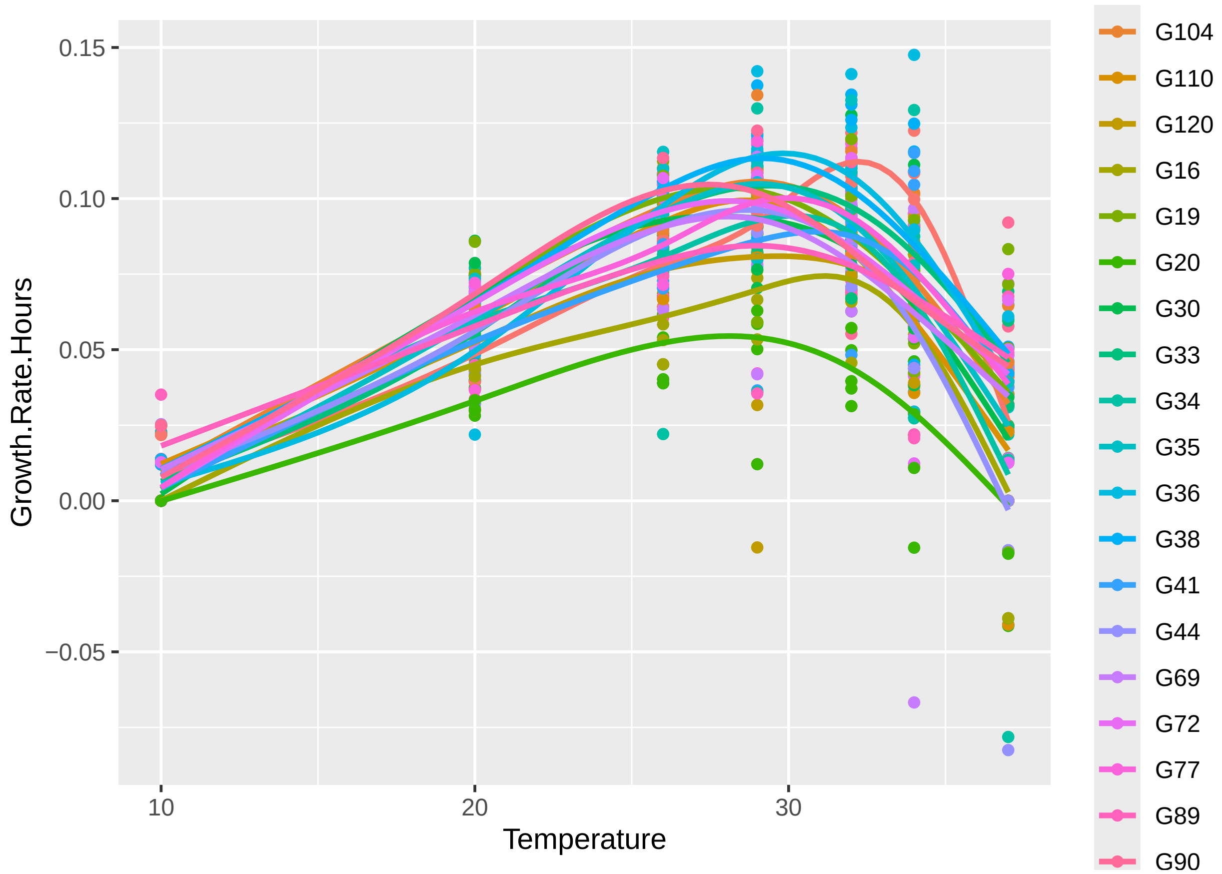
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    428.5      164.6   2.604   0.0180 *
## range_ar       885.4      238.6   3.710   0.0016 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 243.6 on 18 degrees of freedom
## Multiple R-squared:  0.4334, Adjusted R-squared:  0.4019
## F-statistic: 13.77 on 1 and 18 DF,  p-value: 0.001601
```

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 these variables. 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 their 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 predictions from GAMs for each of the phenotypes as the 'observed' phenotype of the outcrossed line at each temperature.

```r
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)
```



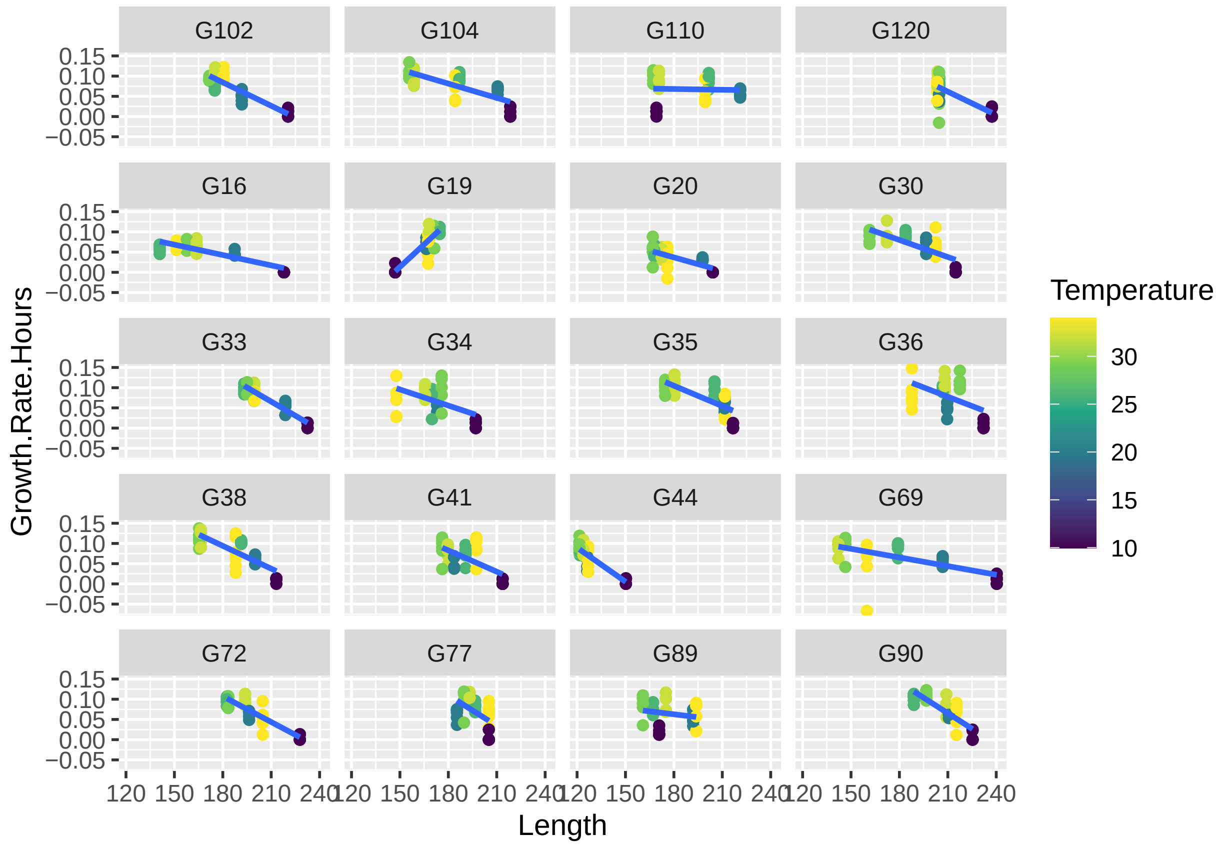

```
### set up a phenotype data frame to join with the growth dataframe

phenotype_combine <- left_join(filter(newdata_length, Temperature %in% c(10,20,26,29,32,34)),
                               filter(newdata_width, Temperature %in% c(10,20,26,29,32,34)), by = c('Li

phenotype_combine <- left_join(phenotype_combine,
                               filter(newdata_ar, Temperature %in% c(10,20,26,29,32,34)), by = c('Line'

phenotype_combine <- left_join(phenotype_combine,
                               filter(newdata_speed, Temperature %in% c(10,20,26,29,32,34)), by = c('Li

colnames(phenotype_combine)[3:6] <- c('Length', 'Width', 'ar', 'Speed')

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

growth_phenotypes <- left_join(growth_combine, phenotype_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 = Length, 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_length_growth <- lm(Growth.Rate.Hours ~ Length*Line, data = growth_phenotypes)

summary(fit_length_growth)
```

```
##
## Call:
## lm(formula = Growth.Rate.Hours ~ Length * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.146482 -0.013695  0.000956  0.016804  0.076025
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        4.330e-01  4.741e-02   9.132  < 2e-16 ***
## Length            -1.936e-03  2.542e-04  -7.615 8.88e-14 ***
## LineG104          -1.403e-01  5.835e-02  -2.405 0.016456 *
## LineG110          -3.535e-01  6.181e-02  -5.719 1.61e-08 ***
## LineG120           3.183e-02  9.202e-02   0.346 0.729565
## LineG16           -2.363e-01  5.529e-02  -4.274 2.20e-05 ***
## LineG19           -9.761e-01  9.329e-02 -10.463  < 2e-16 ***
## LineG20           -1.946e-01  7.109e-02  -2.737 0.006357 **
## LineG30           -9.991e-02  6.511e-02  -1.535 0.125359
## LineG33            1.211e-01  7.787e-02   1.555 0.120473
## LineG34           -1.381e-01  6.928e-02  -1.993 0.046628 *
```

```
## LineG35          -2.546e-02  7.068e-02  -0.360 0.718778
## LineG36          -3.231e-02  8.282e-02  -0.390 0.696565
## LineG38          -5.303e-04  6.662e-02  -0.008 0.993651
## LineG41          -3.404e-02  8.017e-02  -0.425 0.671240
## LineG44          -7.522e-03  7.273e-02  -0.103 0.917667
## LineG69          -2.394e-01  5.240e-02  -4.568 5.85e-06 ***
## LineG72           5.512e-02  7.295e-02   0.756 0.450115
## LineG77           1.291e-01  1.226e-01   1.052 0.292975
## LineG89          -2.825e-01  7.755e-02  -3.643 0.000290 ***
## LineG90           1.694e-01  8.823e-02   1.920 0.055296 .
## Length:LineG104   7.619e-04  3.124e-04   2.439 0.014993 *
## Length:LineG110   1.872e-03  3.294e-04   5.685 1.94e-08 ***
## Length:LineG120   1.542e-05  4.539e-04   0.034 0.972905
## Length:LineG16    1.081e-03  3.034e-04   3.563 0.000393 ***
## Length:LineG19    5.645e-03  5.466e-04  10.329  < 2e-16 ***
## Length:LineG20    8.163e-04  3.871e-04   2.109 0.035309 *
## Length:LineG30    5.296e-04  3.465e-04   1.528 0.126911
## Length:LineG33   -3.906e-04  3.920e-04  -0.996 0.319412
## Length:LineG34    6.073e-04  3.882e-04   1.564 0.118242
## Length:LineG35    2.543e-04  3.645e-04   0.698 0.485550
## Length:LineG36    3.973e-04  4.104e-04   0.968 0.333389
## Length:LineG38    5.541e-05  3.556e-04   0.156 0.876221
## Length:LineG41    1.797e-04  4.239e-04   0.424 0.671794
## Length:LineG44   -8.637e-04  4.979e-04  -1.735 0.083248 .
## Length:LineG69    1.223e-03  2.820e-04   4.337 1.67e-05 ***
## Length:LineG72   -1.808e-04  3.774e-04  -0.479 0.632161
## Length:LineG77   -5.759e-04  6.306e-04  -0.913 0.361424
## Length:LineG89    1.449e-03  4.299e-04   3.370 0.000794 ***
## Length:LineG90   -6.213e-04  4.388e-04  -1.416 0.157275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0256 on 676 degrees of freedom
## Multiple R-squared:  0.5395, Adjusted R-squared:  0.513
## F-statistic: 20.31 on 39 and 676 DF,  p-value: < 2.2e-16
```

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

AIC(fit_length_growth, fit_length_growth_noint)
```

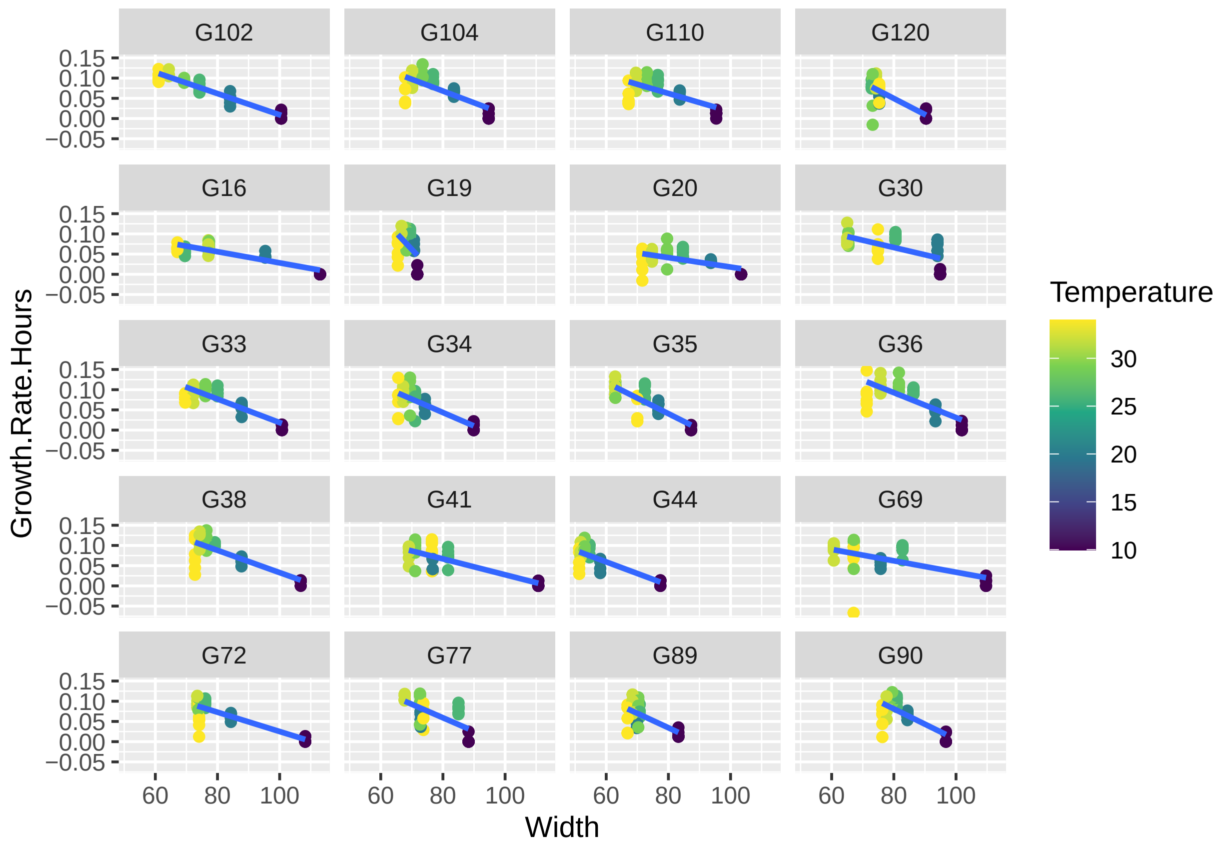
```
##                         df       AIC
## fit_length_growth       41 -3175.531
## fit_length_growth_noint 22 -2993.544
```

In general, we see that there is a negative relationship between length and growth rates across temeperatures 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

```
ggplot(data = growth_phenotypes, aes(x = Width, y = Growth.Rate.Hours)) + geom_point(aes(color = Tempera
  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 ~ Width*Line, data = growth_phenotypes)

summary(fit_width_growth)
```

```
##
## Call:
## lm(formula = Growth.Rate.Hours ~ Width * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.147036 -0.012375  0.002371  0.015232  0.054551
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.2719826  0.0235970  11.526  < 2e-16 ***
## Width           -0.0026278  0.0003075  -8.547  < 2e-16 ***
## LineG104         0.0285944  0.0424855   0.673 0.501153
## LineG110        -0.0292046  0.0410686  -0.711 0.477257
## LineG120         0.0959581  0.0611358   1.570 0.116979
## LineG16         -0.1053475  0.0320162  -3.290 0.001052 **
## LineG19          0.3602263  0.1322516   2.724 0.006620 **
## LineG20         -0.1373458  0.0396220  -3.466 0.000561 ***
## LineG30         -0.0613621  0.0359808  -1.705 0.088577 .
## LineG33          0.0382117  0.0396627   0.963 0.335684
## LineG34          0.0373597  0.0441423   0.846 0.397659
```

```
## LineG35          0.0776195  0.0425087   1.826 0.068296 .
## LineG36          0.0673964  0.0413950   1.628 0.103963
## LineG38          0.0363973  0.0376103   0.968 0.333517
## LineG41         -0.0479572  0.0338242  -1.418 0.156699
## LineG44         -0.0407614  0.0354034  -1.151 0.249998
## LineG69         -0.0971062  0.0309693  -3.136 0.001790 **
## LineG72         -0.0079719  0.0360684  -0.221 0.825143
## LineG77          0.0544055  0.0491178   1.108 0.268405
## LineG89          0.0477134  0.0598159   0.798 0.425342
## LineG90          0.1121073  0.0551418   2.033 0.042435 *
## Width:LineG104 -0.0002783  0.0005461  -0.510 0.610449
## Width:LineG110  0.0003700  0.0005286   0.700 0.484204
## Width:LineG120 -0.0013500  0.0007948  -1.699 0.089859 .
## Width:LineG16   0.0012443  0.0003996   3.114 0.001926 **
## Width:LineG19  -0.0055247  0.0019176  -2.881 0.004088 **
## Width:LineG20   0.0014614  0.0004834   3.023 0.002597 **
## Width:LineG30   0.0008244  0.0004579   1.800 0.072237 .
## Width:LineG33  -0.0002886  0.0004965  -0.581 0.561331
## Width:LineG34  -0.0006964  0.0005944  -1.172 0.241777
## Width:LineG35  -0.0012344  0.0005762  -2.142 0.032529 *
## Width:LineG36  -0.0004564  0.0005022  -0.909 0.363735
## Width:LineG38  -0.0001277  0.0004659  -0.274 0.784079
## Width:LineG41   0.0006627  0.0004262   1.555 0.120426
## Width:LineG44  -0.0002373  0.0005465  -0.434 0.664313
## Width:LineG69   0.0012170  0.0003991   3.050 0.002381 **
## Width:LineG72   0.0002385  0.0004512   0.529 0.597304
## Width:LineG77  -0.0007133  0.0006381  -1.118 0.264038
## Width:LineG89  -0.0009384  0.0008252  -1.137 0.255848
## Width:LineG90  -0.0011592  0.0006754  -1.716 0.086580 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02465 on 676 degrees of freedom
## Multiple R-squared:  0.5733, Adjusted R-squared:  0.5487
## F-statistic: 23.29 on 39 and 676 DF,  p-value: < 2.2e-16
```

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

AIC(fit_width_growth, fit_width_growth_noint)
```

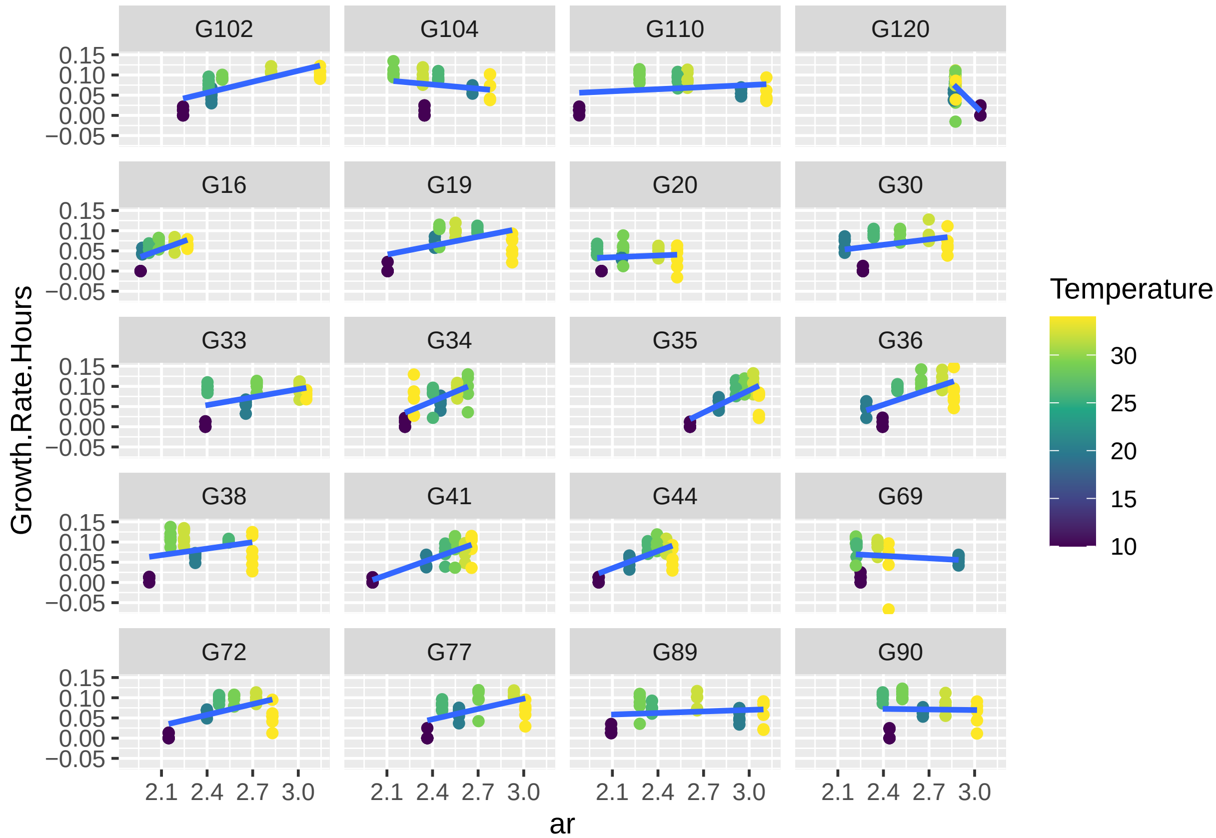
```
##                        df       AIC
## fit_width_growth       41 -3230.017
## fit_width_growth_noint 22 -3176.833
```

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

```r
ggplot(data = growth_phenotypes, aes(x = ar, y = Growth.Rate.Hours)) + geom_point(aes(color = Temperatu
  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 ~ ar*Line, data = growth_phenotypes)

summary(fit_ar_growth)
```

```
##
## Call:
## lm(formula = Growth.Rate.Hours ~ ar * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.132001 -0.019975  0.005835  0.022783  0.085289
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.161187   0.045013  -3.581 0.000367 ***
## ar           0.090542   0.017258   5.247 2.08e-07 ***
## LineG104     0.319014   0.075257   4.239 2.56e-05 ***
## LineG110     0.184939   0.055873   3.310 0.000983 ***
## LineG120     1.299371   0.269035   4.830 1.69e-06 ***
## LineG16     -0.071993   0.106191  -0.678 0.498031
## LineG19      0.047833   0.068936   0.694 0.487995
## LineG20      0.164457   0.075653   2.174 0.030064 *
## LineG30      0.117866   0.070261   1.678 0.093900 .
## LineG33      0.057054   0.070586   0.808 0.419206
## LineG34     -0.150984   0.097769  -1.544 0.122985
## LineG35     -0.297433   0.108259  -2.747 0.006167 **
```

```
## LineG36     -0.084855   0.079367  -1.069 0.285391
## LineG38      0.117095   0.069923   1.675 0.094471 .
## LineG41     -0.101152   0.073858  -1.370 0.171287
## LineG44     -0.101282   0.086286  -1.174 0.240891
## LineG69      0.276174   0.069698   3.962 8.21e-05 ***
## LineG72      0.005739   0.074645   0.077 0.938735
## LineG77      0.003549   0.074599   0.048 0.962068
## LineG89      0.192917   0.058708   3.286 0.001069 **
## LineG90      0.244986   0.077864   3.146 0.001726 **
## ar:LineG104 -0.124484   0.029982  -4.152 3.72e-05 ***
## ar:LineG110 -0.073388   0.021475  -3.417 0.000670 ***
## ar:LineG120 -0.461643   0.093121  -4.957 9.04e-07 ***
## ar:LineG16   0.046194   0.049250   0.938 0.348610
## ar:LineG19  -0.017067   0.026869  -0.635 0.525504
## ar:LineG20  -0.075785   0.032339  -2.343 0.019394 *
## ar:LineG30  -0.045366   0.027808  -1.631 0.103271
## ar:LineG33  -0.024805   0.026418  -0.939 0.348094
## ar:LineG34   0.065878   0.039685   1.660 0.097375 .
## ar:LineG35   0.092345   0.038070   2.426 0.015540 *
## ar:LineG36   0.034772   0.030598   1.136 0.256189
## ar:LineG38  -0.037276   0.028628  -1.302 0.193329
## ar:LineG41   0.043362   0.029447   1.473 0.141337
## ar:LineG44   0.051210   0.036093   1.419 0.156406
## ar:LineG69  -0.110978   0.028035  -3.959 8.34e-05 ***
## ar:LineG72  -0.001733   0.029143  -0.059 0.952611
## ar:LineG77  -0.005387   0.028081  -0.192 0.847912
## ar:LineG89  -0.077816   0.022556  -3.450 0.000596 ***
## ar:LineG90  -0.095264   0.029544  -3.225 0.001323 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03126 on 676 degrees of freedom
## Multiple R-squared:  0.3134, Adjusted R-squared:  0.2738
## F-statistic: 7.913 on 39 and 676 DF,  p-value: < 2.2e-16
```

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

AIC(fit_ar_growth, fit_ar_growth_noint)
```

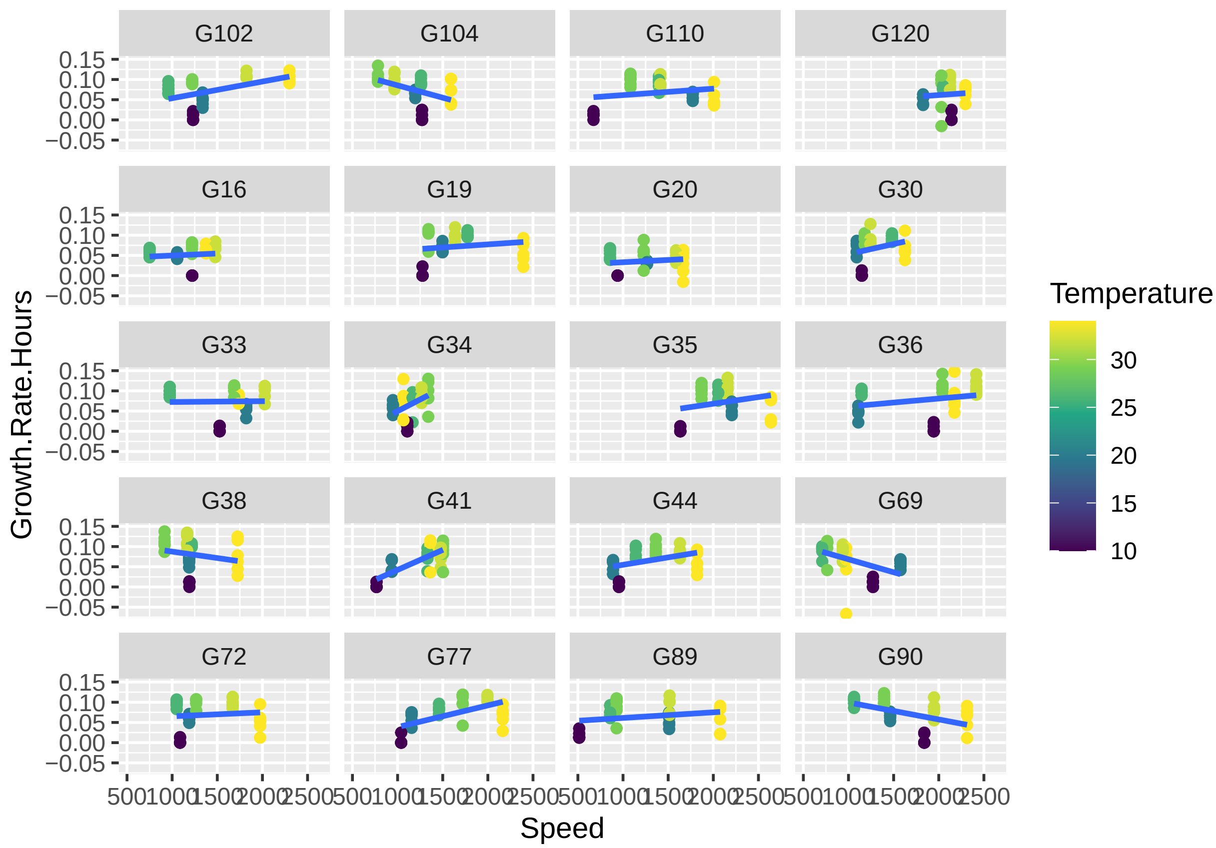
```
##                     df       AIC
## fit_ar_growth       41 -2889.492
## fit_ar_growth_noint 22 -2798.947
```

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 = Speed, y = Growth.Rate.Hours)) + geom_point(aes(color = Tempera
  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 ~ Speed*Line, data = growth_phenotypes)

summary(fit_speed_growth)
```

```
##
## Call:
## lm(formula = Growth.Rate.Hours ~ Speed * Line, data = growth_phenotypes)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.136894 -0.019464  0.007786  0.025283  0.071941
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.225e-02  1.946e-02   0.630  0.52920
## Speed            4.134e-05  1.259e-05   3.283  0.00108 **
## LineG104         1.343e-01  3.310e-02   4.057 5.55e-05 ***
## LineG110         3.321e-02  2.714e-02   1.224  0.22152
## LineG120         1.770e-02  9.001e-02   0.197  0.84415
## LineG16          2.736e-02  3.501e-02   0.782  0.43469
## LineG19          3.481e-02  3.233e-02   1.077  0.28199
## LineG20          9.445e-03  3.119e-02   0.303  0.76214
## LineG30         -8.820e-03  4.299e-02  -0.205  0.83752
## LineG33          5.829e-02  3.454e-02   1.687  0.09199 .
## LineG34         -7.793e-02  5.442e-02  -1.432  0.15265
## LineG35         -9.767e-03  4.327e-02  -0.226  0.82146
```

```
## LineG36         2.913e-02  2.874e-02   1.013  0.31126
## LineG38         1.072e-01  3.523e-02   3.043  0.00243 **
## LineG41        -6.907e-02  3.215e-02  -2.148  0.03203 *
## LineG44         5.954e-03  2.959e-02   0.201  0.84056
## LineG69         1.205e-01  2.816e-02   4.279 2.15e-05 ***
## LineG72         4.296e-02  3.068e-02   1.400  0.16191
## LineG77        -2.752e-02  2.975e-02  -0.925  0.35526
## LineG89         3.505e-02  2.432e-02   1.442  0.14989
## LineG90         1.294e-01  3.067e-02   4.218 2.80e-05 ***
## Speed:LineG104 -1.024e-04  2.554e-05  -4.009 6.79e-05 ***
## Speed:LineG110 -2.541e-05  1.809e-05  -1.405  0.16043
## Speed:LineG120 -2.559e-05  4.394e-05  -0.582  0.56050
## Speed:LineG16  -3.130e-05  2.721e-05  -1.150  0.25046
## Speed:LineG19  -2.626e-05  1.977e-05  -1.329  0.18441
## Speed:LineG20  -3.000e-05  2.267e-05  -1.323  0.18621
## Speed:LineG30   8.467e-06  3.188e-05   0.266  0.79060
## Speed:LineG33  -3.938e-05  2.130e-05  -1.849  0.06489 .
## Speed:LineG34   7.420e-05  4.574e-05   1.622  0.10526
## Speed:LineG35  -8.481e-06  2.218e-05  -0.382  0.70232
## Speed:LineG36  -2.154e-05  1.692e-05  -1.273  0.20344
## Speed:LineG38  -7.323e-05  2.653e-05  -2.760  0.00593 **
## Speed:LineG41   5.771e-05  2.388e-05   2.416  0.01595 *
## Speed:LineG44  -4.666e-06  2.082e-05  -0.224  0.82276
## Speed:LineG69  -1.056e-04  2.265e-05  -4.662 3.77e-06 ***
## Speed:LineG72  -3.140e-05  2.098e-05  -1.496  0.13502
## Speed:LineG77   1.242e-05  1.862e-05   0.667  0.50483
## Speed:LineG89  -2.746e-05  1.665e-05  -1.649  0.09956 .
## Speed:LineG90  -8.343e-05  1.867e-05  -4.468 9.24e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03392 on 676 degrees of freedom
## Multiple R-squared:  0.1919, Adjusted R-squared:  0.1452
## F-statistic: 4.115 on 39 and 676 DF,  p-value: 1.396e-14
```

```
fit_speed_growth_noint <- lm(Growth.Rate.Hours ~ 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 -2772.772
## fit_speed_growth_noint 22 -2720.915
## 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 influences 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, we will fit GAMs to the growth data and then use the predictions from the GAMs to calculate the difference between the maximum and minimum growth rate over the temperature range for which we have growth data. We can then ask whether plasticity in any of the phenotypes predicts the extent of plasticity in growth rates.