Post-hocs and planned comparisons

KEY

Wed Mar 10 13:48:51 2021

Import and summary of data

We will work eel larval development data. Import the “eels.csv” into a data frame called “eels”.

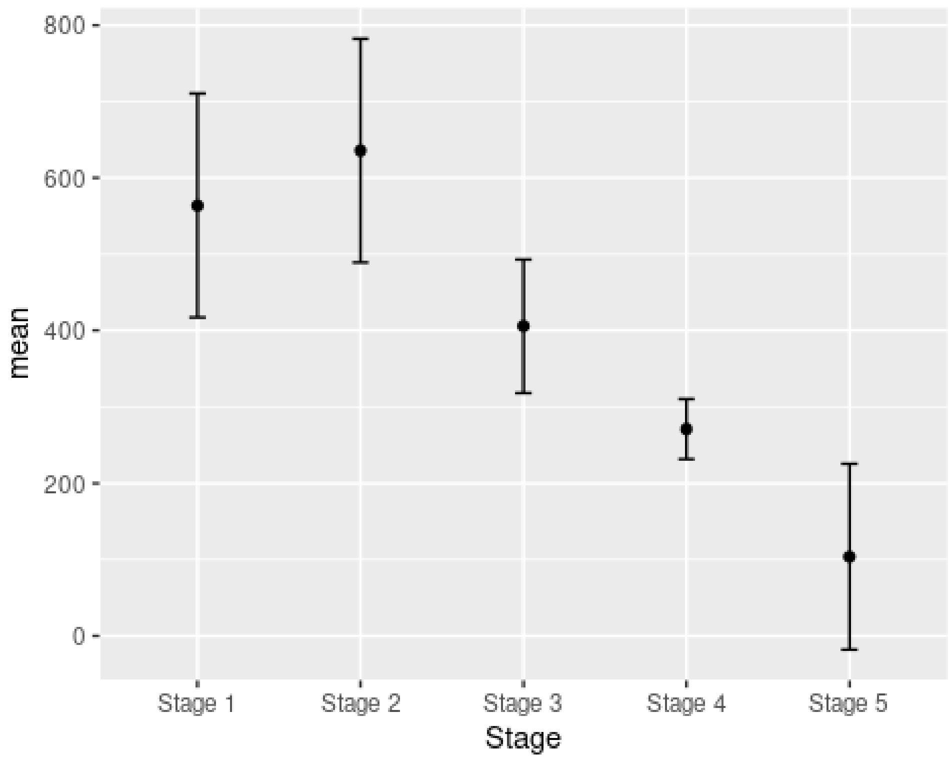
```
eels <- read.csv("eels.csv")
 eels$Stage <- as.factor(eels$Stage)
```

To plot the means and 95% confidence intervals use summarySE() to get the summary statistics you need:

```
library(biol531)
summarySE(eels, "HA","Stage") -> eels.sumstats
```

Then use the summary data set you created to plot the means and 95% confidence intervals using ggplot2:

```
library(ggplot2)
ggplot(eels.sumstats, aes(x = Stage, y = mean)) + geom_errorbar(aes(ymin =
lower, ymax = upper), width = 0.1) + geom_point()
```



Question: the stages are in their proper order on the plot. Stage is clearly an ordinal variable, but does that mean that R is considering Stage to be ordinal?

> No, the categories are in alphabetical order, but the factor is not considered ordered at this point. The ordering will not be used in this first analysis.

Fitting the linear model

Now run the lm() command to fit the model, using HA as the response variable and Stage as the predictor:

```
eels.ha.lm <- lm(HA ~ Stage, data = eels)
anova(eels.ha.lm)

## Analysis of Variance Table
##
## Response: HA
##           Df Sum Sq Mean Sq F value    Pr(>F)
## Stage      4 747378  186844  35.331 1.805e-07 ***
## Residuals 15  79327    5288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Question: is the Stage term significant? How do you know?

> Yes, because the p-value for Stage is significant.

Question: can you tell at this point which stages differ?

No, only that there are at least two Stages have mean HA that differ.

Load the multcomp library:

```
library(multcomp)

## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':
##
##     geyser
```

Run the glht() command, from multcomp, for the Tukey procedure:

```
eels.ha.tukey <- glht(eels.ha.lm, linfct = mcp(Stage = "Tukey"))
summary(eels.ha.tukey)

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = HA ~ Stage, data = eels)
##
## Linear Hypotheses:
##                        Estimate Std. Error t value Pr(>|t|)
## Stage 2 - Stage 1 == 0    71.86      51.42   1.397  0.63840
## Stage 3 - Stage 1 == 0  -158.10      51.42  -3.075  0.05127 .
## Stage 4 - Stage 1 == 0  -292.84      51.42  -5.695  < 0.001 ***
## Stage 5 - Stage 1 == 0  -460.25      51.42  -8.950  < 0.001 ***
## Stage 3 - Stage 2 == 0  -229.96      51.42  -4.472  0.00347 **
## Stage 4 - Stage 2 == 0  -364.69      51.42  -7.092  < 0.001 ***
## Stage 5 - Stage 2 == 0  -532.10      51.42 -10.348  < 0.001 ***
## Stage 4 - Stage 3 == 0  -134.73      51.42  -2.620  0.11624
## Stage 5 - Stage 3 == 0  -302.14      51.42  -5.876  < 0.001 ***
## Stage 5 - Stage 4 == 0  -167.41      51.42  -3.256  0.03648 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

cld(eels.ha.tukey)
```

```
## Stage 1 Stage 2 Stage 3 Stage 4 Stage 5
##    "cd"     "d"    "bc"     "b"     "a"
```

Question: which stages are different from one another?

Working from first to last stage, Stage 1 is different from Stage 4 and 5, Stage 2 is different from Stage 3, 4, and 5. Stage 3 is different from Stage 5. STage 4 is different from Stage 5.

Dunnett's comparisons are all groups against one. Usually the one that is compared against is a "control" group, and all the others are treatments of various kinds. We don't have data like that, but since Stage 1 is set as the baseline group we can use Dunnett's test to compare all of the other stages against Stage 1:

```
eels.ha.dunnett <- glht(eels.ha.lm, linfct = mcp(Stage = "Dunnett"))
summary(eels.ha.dunnett)

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Dunnett Contrasts
##
##
## Fit: lm(formula = HA ~ Stage, data = eels)
##
## Linear Hypotheses:
##                        Estimate Std. Error t value Pr(>|t|)
## Stage 2 - Stage 1 == 0    71.86      51.42   1.397   0.4596
## Stage 3 - Stage 1 == 0  -158.10      51.42  -3.075   0.0254 *
## Stage 4 - Stage 1 == 0  -292.84      51.42  -5.695   <0.001 ***
## Stage 5 - Stage 1 == 0  -460.25      51.42  -8.950   <0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Question: are all of the stages different from Stage 1?

No, Stage 2 is not, but the rest of the stages are different from Stage 1.

Post-hoc comparisons of your choice

We need a matrix that defines comparisons between sequential stages.

```
s1vs2 <- c(1,-1,0,0,0)
s2vs3 <- c(0,1,-1,0,0)
s3vs4 <- c(0,0,1,-1,0)
s4vs5 <- c(0,0,0,1,-1)

rbind(s1vs2, s2vs3, s3vs4, s4vs5) -> seq.comparisons
```

Now get the multiple comparisons for this set of sequential comparisons of means:

```
glht(eels.ha.lm, linfct = mcp(Stage = seq.comparisons)) ->
seq.comparisons.glht
summary(seq.comparisons.glht)

## 
##   Simultaneous Tests for General Linear Hypotheses
## 
## Multiple Comparisons of Means: User-defined Contrasts
## 
## 
## Fit: lm(formula = HA ~ Stage, data = eels)
## 
## Linear Hypotheses:
##              Estimate Std. Error t value Pr(>|t|)   
## s1vs2 == 0   -71.86      51.42  -1.397  0.48202   
## s2vs3 == 0   229.96      51.42   4.472  0.00165 **
## s3vs4 == 0   134.73      51.42   2.620  0.06526 . 
## s4vs5 == 0   167.41      51.42   3.256  0.01878 * 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Question: which adjacent means are different?

Stage 2 is different from stage 3, and 4 is different from 5.

Orthogonal contrasts

We will use the second set of possible orthogonal contrasts, which compares each stage against the mean of those that follow.

First, make four vectors that have the weights shown in each column of the Set 2 contrast matrix:

```
Contrast.1 <- c(4,-1,-1,-1,-1)
Contrast.2 <- c(0,3,-1,-1,-1)
Contrast.3 <- c(0,0,2,-1,-1)
Contrast.4 <- c(0,0,0,1,-1)
```

Bind the four vectors of contrast weights into a single contrast matrix:

```
cbind(Contrast.1, Contrast.2, Contrast.3, Contrast.4) -> set.2.contrasts
```

Make a copy of the Stage variable, and call it Stage.custom so that we can assign these custom contrasts to it:

```
eels$Stage.custom <- eels$Stage
```

Assign the hand-made custom contrasts to the Stage variable in eels:

```
contrasts(eels$Stage.custom) <- set.2.contrasts
```

Now we can fit a model using Stage.custom as a predictor and HA as a response, just as we did before, but the coefficient tests will be tests of these four contrasts:

```
eels.ha.custom.contrasts.lm <- lm(HA ~ Stage.custom, data = eels)
summary(eels.ha.custom.contrasts.lm)

##
## Call:
## lm(formula = HA ~ Stage.custom, data = eels)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.94  -43.12   12.68   30.44  122.39
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                395.938     16.261  24.349 1.79e-13 ***
## Stage.customContrast.1      41.966      8.131   5.162 0.000116 ***
## Stage.customContrast.2      93.896     10.497   8.945 2.12e-07 ***
## Stage.customContrast.3      72.813     14.844   4.905 0.000190 ***
## Stage.customContrast.4      83.705     25.711   3.256 0.005322 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.72 on 15 degrees of freedom
## Multiple R-squared:  0.904,  Adjusted R-squared:  0.8785
## F-statistic: 35.33 on 4 and 15 DF,  p-value: 1.805e-07
```

Orthogonal polynomials - analyzing ordinal trends

R uses polynomial contrasts by default for ordered factors. If we make a version of the Stage variable that is an ordered factor, and then use it in a linear model, R will give us tests of the ordinal trends across the five stages.

Make a copy of Stage as an ordered factor:

```
eels$Stage.ordered <- factor(eels$Stage, ordered = T)
```

To see what the polynomial weights look like, make a set of orthogonal polynomial weights and display them:

```
contr.poly(5) -> poly.wts
poly.wts

##              .L         .Q            .C         ^4
## [1,] -0.6324555  0.5345225 -3.162278e-01  0.1195229
## [2,] -0.3162278 -0.2672612  6.324555e-01 -0.4780914
## [3,]  0.0000000 -0.5345225 -4.095972e-16  0.7171372
## [4,]  0.3162278 -0.2672612 -6.324555e-01 -0.4780914
## [5,]  0.6324555  0.5345225  3.162278e-01  0.1195229
```

Question: these weights describe a pattern of change from first to last stage. How? Use Linear and Quadratic weights as an example.

> Linear weights start at a negative value and increase by an equal amount across the levels. The quadratic term is a parabola that opens upward, so it starts at positive values, declines to a minimum, and then increases back to the same value as it started.

Fit a model using the ordered factor version of stage:

```
eels.op.lm <- lm(HA ~ Stage.ordered, data = eels)
summary(eels.op.lm)

##
## Call:
## lm(formula = HA ~ Stage.ordered, data = eels)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -116.94  -43.12   12.68   30.44  122.39
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       395.94      16.26  24.349 1.79e-13 ***
## Stage.ordered.L  -406.41      36.36 -11.177 1.13e-08 ***
## Stage.ordered.Q  -102.44      36.36  -2.817   0.0130 *
## Stage.ordered.C    85.11      36.36   2.341   0.0335 *
## Stage.ordered^4   -62.74      36.36  -1.726   0.1050
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 72.72 on 15 degrees of freedom
## Multiple R-squared:  0.904,  Adjusted R-squared:  0.8785
## F-statistic: 35.33 on 4 and 15 DF,  p-value: 1.805e-07
```

Get the ANOVA table - it doesn't look any different:

```
anova(eels.op.lm)

## Analysis of Variance Table
##
## Response: HA
##               Df Sum Sq Mean Sq F value    Pr(>F)
## Stage.ordered  4 747378  186844  35.331 1.805e-07 ***
## Residuals     15  79327    5288
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Run Tukey tests on this ordinal stage variable:

```
summary(glht(eels.op.lm, linfct = mcp(Stage.ordered = "Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lm(formula = HA ~ Stage.ordered, data = eels)
##
## Linear Hypotheses:
##                        Estimate Std. Error t value Pr(>|t|)    
## Stage 2 - Stage 1 == 0    71.86      51.42   1.397  0.63840    
## Stage 3 - Stage 1 == 0  -158.10      51.42  -3.075  0.05126 .  
## Stage 4 - Stage 1 == 0  -292.83      51.42  -5.695  < 0.001 ***
## Stage 5 - Stage 1 == 0  -460.25      51.42  -8.950  < 0.001 ***
## Stage 3 - Stage 2 == 0  -229.96      51.42  -4.472  0.00347 ** 
## Stage 4 - Stage 2 == 0  -364.69      51.42  -7.092  < 0.001 ***
## Stage 5 - Stage 2 == 0  -532.10      51.42 -10.348  < 0.001 ***
## Stage 4 - Stage 3 == 0  -134.73      51.42  -2.620  0.11620    
## Stage 5 - Stage 3 == 0  -302.14      51.42  -5.876  < 0.001 ***
## Stage 5 - Stage 4 == 0  -167.41      51.42  -3.256  0.03656 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Question: are there any changes to these Tukey tests compared with the analysis with the unordered factor you did previously?

No, the comparisons of means are all the same.

The tests of polynomial trends are seen in the coefficient tests:

```
summary(eels.op.lm)

##
## Call:
## lm(formula = HA ~ Stage.ordered, data = eels)
##
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -116.94  -43.12   12.68   30.44  122.39 
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        395.94      16.26  24.349 1.79e-13 ***
## Stage.ordered.L   -406.41      36.36 -11.177 1.13e-08 ***
## Stage.ordered.Q   -102.44      36.36  -2.817   0.0130 *  
## Stage.ordered.C     85.11      36.36   2.341   0.0335 *  
## Stage.ordered^4    -62.74      36.36  -1.726   0.1050    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 72.72 on 15 degrees of freedom
## Multiple R-squared:  0.904,  Adjusted R-squared:  0.8785
## F-statistic: 35.33 on 4 and 15 DF,  p-value: 1.805e-07
```

Question: which polynomial trends were statistically significant?

Linear, Quadratic, and Cubic trends were significant.

Corrections for multiple p-values

We just completed an analysis of HA, but the data set includes measures of change in water and in neutral sugar as well. Fit a model using water as a response, and Stage as a predictor, then get the summary output:

```
lm(water ~ Stage, data = eels) -> water.lm
summary(water.lm)

##
## Call:
## lm(formula = water ~ Stage, data = eels)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9475 -1.9519 -0.8325  1.8881  6.6650
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    90.937      1.769  51.396  < 2e-16 ***
## StageStage 2   -3.100      2.502  -1.239  0.23442
## StageStage 3   -2.602      2.502  -1.040  0.31478
## StageStage 4   -5.447      2.502  -2.177  0.04586 *
## StageStage 5   -9.915      2.502  -3.962  0.00125 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.539 on 15 degrees of freedom
## Multiple R-squared:  0.5422, Adjusted R-squared:  0.4201
## F-statistic: 4.442 on 4 and 15 DF,  p-value: 0.01445
```

Fit a model using NS as a response, and get the summary output:

```
lm(NS ~ Stage, data = eels) -> ns.lm
summary(ns.lm)

##
## Call:
## lm(formula = NS ~ Stage, data = eels)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2575 -1.0269  0.2263  0.7138  2.4450
```

```
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      4.7075     0.8413   5.595 5.11e-05 ***
## StageStage 2     3.9275     1.1898   3.301 0.004851 **
## StageStage 3     4.8900     1.1898   4.110 0.000928 ***
## StageStage 4     9.1325     1.1898   7.675 1.43e-06 ***
## StageStage 5    13.1975     1.1898  11.092 1.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.683 on 15 degrees of freedom
## Multiple R-squared:  0.9064, Adjusted R-squared:  0.8815
## F-statistic: 36.33 on 4 and 15 DF,  p-value: 1.496e-07
```

Question: according to the model-level omnibus test (i.e. the p-value at the bottom of the summary output), do NS and water change during development? How do you know?

Yes they do, both have p-values less than 0.05.

Calculate the adjusted alpha levels:

```
0.05/3

## [1] 0.01666667

1 - (1 - 0.05)^(1/3)

## [1] 0.01695243
```

Question: are any of the variables no longer statistically significant at either of these adjusted alpha levels?

All are still statistically significant after adjustment, but water is much closer to the adjusted alpha level.

OPTIONAL for the adventurous

Three of the trends were statistically significant. To understand better how the polynomial coefficients work, and what patterns are being tested by each of the trends, we can predict the stage means using the linear trend, then add the quadratic trend, and then the cubic. You'll see that we can only approximately predict the means of stages with the linear trend, but the prediction gets better each time we add another one. Just like we needed four dummy coded columns to reproduce a factor with five stages, we can reproduce the five stage means if we included all four polynomial weights, but the fourth order trend was not significant. We will see why this was the case.

First, get the coefficients:

```
coef(eels.op.lm) -> stage.coef
stage.coef

##     (Intercept) Stage.ordered.L Stage.ordered.Q Stage.ordered.C 
Stage.ordered^4 
##       395.93750      -406.41039      -102.44324        85.10954        -
62.74353
```

Get the contrast weights:

```
contrasts(eels$Stage.ordered) -> stage.contrasts
stage.contrasts

##              .L         .Q            .C         ^4
## [1,] -0.6324555  0.5345225 -3.162278e-01  0.1195229
## [2,] -0.3162278 -0.2672612  6.324555e-01 -0.4780914
## [3,]  0.0000000 -0.5345225 -4.095972e-16  0.7171372
## [4,]  0.3162278 -0.2672612 -6.324555e-01 -0.4780914
## [5,]  0.6324555  0.5345225  3.162278e-01  0.1195229
```

Calculate each of the three significant trends - linear first, then add the quadratic, then add the cubic:

```
stage.coef[1] + stage.coef[2]*stage.contrasts[,1] -> linear
linear + stage.coef[3]*stage.contrasts[,2] -> quadratic
quadratic + stage.coef[4]*stage.contrasts[,3] -> cubic
```

We need the trends stacked for graphing - make a data frame from the three statistically significant trends, and then stack the data frame:

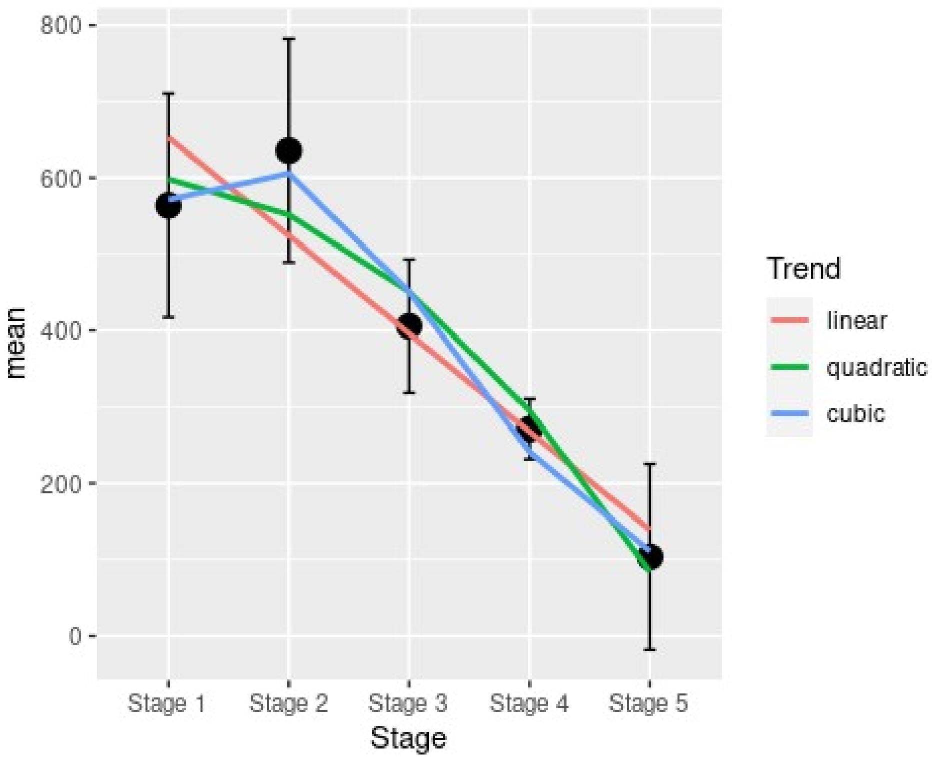
```
data.frame(linear, quadratic, cubic) -> trends
stack(trends) -> trends.stacked
```

Add stage names to the data frame, and re-name the columns:

```
data.frame(levels(eels$Stage.ordered), trends.stacked) -> trends.stacked
names(trends.stacked) <- c("Stage","mean","Trend")
```

Plot the observed means and confidence intervals along with the three trends:

```
ggplot(eels.sumstats, aes(x = Stage, y = mean)) + geom_errorbar(aes(ymin = 
lower, ymax = upper), width = 0.1) + geom_point(size = 4) + 
geom_line(trends.stacked, mapping = aes(x = Stage, y = mean, color = Trend, 
group = Trend), size = 1)
```


800
600
400
200
0
mean
Stage 1
Stage 2
Stage 3
Stage 4
Stage 5
Stage
Trend
linear
quadratic
cubic