

HW4. T-tests and ANOVA

Linguistic data: Quantitative analysis and vizualization

Universal linguistic hierarchies: a case of Modern Greek (Standard and Cypriot dialects)

Data (responces (<https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/greek-word-order-mono-acceptability-coded-rt.txt>), questionnaire (https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/greek-word-order-mono_socio.txt)) adapted from the survey: Leivada, Evelina; Westergaard, Marit, 2019, Universal linguistic hierarchies are not innately wired (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6679903/#fn-1>). PeerJ, v.7.

Source of data: TROLLing repository: Leivada, Evelina; Westergaard, Marit, 2019, “Replication Data for: Universal linguistic hierarchies are not innately wired”, <https://doi.org/10.18710/NTLLUF> (<https://doi.org/10.18710/NTLLUF>), DataverseNO, V1

Constructions with two adjectives

In English, the order of two adjectives in phrases like:

```
a big black bag # ok
*a black big bag # unacceptable, ungrammatically ill-formed, or semantically anomalous
```

is powered by the semantic class of adjective (e.g. the `color` adjective closer to the noun than the `size` adjective).

A syntactic hierarchy of closeness to the noun in Chomsky's Universal Grammar suggests the following order and is claimed to be innate and universal (= valid for all languages).

```
Subjective Comment > Evidential > Size > Length
> Height > Speed > Depth > Width > Temperature > Wetness > Age
> Shape > Color > Nationality/Origin > Material
# (adapted from Scott, 2002: 114)
```

The goal of Leivada & Westergaard research is identify what happens when people process orderings that either comply with the hierarchy or violate it.

Method

In the first experiment, 140 neurotypical, adult speakers completed a timed forced choice task that featured stimuli showing a combination of two adjectives and a concrete noun (e.g., *I bought a square black table*). Two types of responses were collected:

- i. acceptability judgments on a 3-point Likert scale that featured the options
 1. wrong,
 2. neither correct nor wrong,
 3. correct;
- ii. reaction times (RT).

The task featured three conditions: 1. size adjective > nationality adjective, 2. color adjective > shape adjective, 3. subjective comment adjective > material adjective. Each condition had two orders. In the congruent order, the adjective pair was ordered in agreement with what is traditionally accepted as dictated by the universal hierarchy. In the incongruent order, the ordering was reversed, thus the hierarchy was violated.

In the second experiment, 30 bidialectals (native speakers of Standard and Cypriot Greek) were tested in both language varieties, 36 observations per participant, 18 for each variety.

Two kinds of fillers (<https://www.hlp.rochester.edu/resources/BCS152-Tutorial/Fillers.html>) were used in both experiments, FillerAcceptable and FillerUnacceptable – sentences that included well-formed and ungrammatical structures, respectively. In both tasks the ratio of fillers to actual test structures was 2:1.

Data

```
suppressMessages(library(tidyverse))
```

```
mono_socio <- read_csv2("https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/greek-word-order-mono_socio.txt")
mono <- read_csv2("https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/greek-word-order-mono-acceptability-coded-rt.txt")
```

see also reading key for the data (<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6679903/bin/peerj-07-7438-s001.txt>)

```
mono_socio
```

```
## # A tibble: 180 x 4
##   ParticipantID      TypeOfQuestion QuestionCategory Responce
##   <chr>             <chr>             <chr>             <chr>
## 1 2bc93481d25426f259a0f1a3d5d3e... Form          age              58
## 2 2bc93481d25426f259a0f1a3d5d3e... Form          country1        Nicosia
## 3 2bc93481d25426f259a0f1a3d5d3e... Form          country2        Kaimakli,Nico...
## 4 2bc93481d25426f259a0f1a3d5d3e... Form          education       College
## 5 2bc93481d25426f259a0f1a3d5d3e... Form          handedness      right
## 6 2bc93481d25426f259a0f1a3d5d3e... Form          sex             female
## 7 4b4c9debd4aedeblfd3dfc9f1cab3... Form          age             24
## 8 4b4c9debd4aedeblfd3dfc9f1cab3... Form          country1        Nicosia
## 9 4b4c9debd4aedeblfd3dfc9f1cab3... Form          country2        Nicosia
## 10 4b4c9debd4aedeblfd3dfc9f1cab3... Form          education       UniversityDeg...
## # ... with 170 more rows
```

1. Data overview

1.1

Use `mono_socio` dataframe to answer the following questions:

1. How many participants are mentioned in this dataframe?
2. How many of them are males and females?
3. Which education levels are mentioned in the dataframe?
4. How many participants of each education levels are present?
5. How many left- and right-randed participants are present?

The following functions from tidyverse can be usefult for this problem: `filter` , `group_by` , `count` and `distinct` . (Another approach is to use `pivot_wider` .)

Compare your overview with that reported in Table 1 of the article. Sometimes replication data provided by authors does not allow on to reproduce their results. Let's look at another dataframe, `mono`, that contains results of experiment 1.

```
mono
```

```
## # A tibble: 7,560 x 6
##   ParticipantID TypeOfQuestion TypeOfStimuli WordOrder ResponseAccepta... RT
##   <chr>          <chr>          <chr>          <chr>          <chr>          <dbl>
## 1 00e0b159cf5b9... AcceptabilityJ... Size-Nationa... Congruent Correct          1591
## 2 00e0b159cf5b9... AcceptabilityJ... Size-Nationa... Incongru... Correct          5504
## 3 00e0b159cf5b9... AcceptabilityJ... Size-Nationa... Congruent Correct          1333
## 4 00e0b159cf5b9... AcceptabilityJ... Size-Nationa... Incongru... Correct          1543
## 5 00e0b159cf5b9... AcceptabilityJ... Size-Nationa... Congruent Correct          1815
## 6 00e0b159cf5b9... AcceptabilityJ... Size-Nationa... Incongru... Correct          3059
## 7 00e0b159cf5b9... AcceptabilityJ... Shape-Color   Congruent Correct          1700
## 8 00e0b159cf5b9... AcceptabilityJ... Shape-Color   Incongru... Correct          2745
## 9 00e0b159cf5b9... AcceptabilityJ... Shape-Color   Congruent Correct          3237
## 10 00e0b159cf5b9... AcceptabilityJ... Shape-Color   Incongru... Correct          2911
## # ... with 7,550 more rows
```

1.2

Create a plot that shows the RT distribution in experiment 1 (all participants and conditions taken together). What kind of plot would you choose? Use `ggplot()` for this problem.

```
library(ggplot2)
```

Can we say that RT approximately follows normal distribution? Which features of RT distribution contradicts this assumption? (E.g. long left tail, long right tail, outliers, skewness, etc.)

1.3

Normalise data applying the logarithm with base 10 ($RT_{log} = \log_{10}(RT)$). Use `mutate`.

1.4

Create a density plot that shows the RT_{log} distribution.

Can we say that RT_{log} approximately follows normal distribution? What features of RT_{log} distribution contradicts this assumption? (E.g. long left tail, long right tail, outliers, skewness, etc.)

1.5

Give a summary of RT_{log} distribution (min, max, mean, median, standard deviation)

```
# hint: sd()
```

1.6

Filter out outliers. Remove from the table the following observations: * responses whose RT is below 600 ms (i.e., when a button is pressed too fast, without allowing enough time for actual consideration of the presented stimuli)

* responses whose RT_{log} deviates from the mean value of RT_{log} for more than 3 standard deviations

* fillers (both acceptable and unacceptable)

Convert relevant variables to factors and save filtered data as `mono1` .

```
# mono1 <-  
# mono %>%  
# filter here  
  
# select(ParticipantID, TypeOfStimuli, WordOrder, AcceptabilityJ =  
# ResponseAcceptabilityJudgement, RTlog) %>%  
# mutate(ParticipantID = as.factor(ParticipantID),  
#        TypeOfStimuli = as.factor(TypeOfStimuli),  
#        WordOrder = as.factor(WordOrder),  
#        AcceptabilityJ = as.factor(AcceptabilityJ))
```

1.7

Calculate the number of observations in `mono1` .

1.8

Reproduce Figure 1 from the article using `ggplot` .

Hint: You can make a summary and use `geom_col()` (see example here (<https://r-graphics.org/recipe-colors-mapping>)). Use either `facet_wrap` or `facet_grid` to make six plots. Note that the figures created in 1.8-1.0 may look different from what plotted in the article.

1.9

Reproduce Figure 2 from the article using `ggplot`.

1.10

Reproduce Figure 7 from the article using `ggplot`.

1.11

For the same data, draw a lineplot for group means and standard errors using `ggline()` :

2. Difference in reaction time

Let us test if there are any differences in the reaction time between congruent and incongruent orders. Reaction time is a numeric variable so we can use t-test to compare means. One option is to use two-sample t-test. However, as we have data for congruent and incongruent orders for *the same participants*, it is better to use *paired t-test* here. In paired t-test, for each participant, we will find the difference of their reaction time in congruent and incongruent orders, and compare these differences with 0 using 1-sample t-test. To make sure that our data satisfy assumptions of t-test (values that we compare are independent samples from some approximately normal distributions), we will find the mean logarithm of reaction time for each participant (across observations in all conditions), and consider them as our new sample.

2.1 Summarising

Use `group_by` and `summarise` to find mean logarithm of reaction time for each participant and each word order. Put this dataframe to `mean_rtlog_long` variable. It should be like

```
# A tibble: 280 x 3
  ParticipantID      WordOrder  RTlog
  <fct>          <fct>      <dbl>
1 00e0b159cf5b9abcc73b92506d8b1c38 Congruent    3.24
2 00e0b159cf5b9abcc73b92506d8b1c38 Incongruent  3.47
3 021a49cde484f8fa18439f026ec99459 Congruent    3.22
4 021a49cde484f8fa18439f026ec99459 Incongruent  3.21
...
```

2.2. Pivoting

Use `pivot_wider` to spread values of `RTlog` in `mean_rtlog_long` into two columns: `Congruent` and `Incongruent`. Put new dataframe in variable `mean_rtlog`. It should look like

```
# A tibble: 140 x 3
  ParticipantID      Congruent Incongruent
  <fct>          <dbl>      <dbl>
1 00e0b159cf5b9abcc73b92506d8b1c38    3.24    3.47
2 021a49cde484f8fa18439f026ec99459    3.22    3.21
3 02810ff2a65eae2b3e54ac57d906309d    3.46    3.36
```

2.3. Two-sample t-test

Let us try to apply two-sample t-test to our data. Consider values in columns `Congruent` and `Incongruent` as two independent samples. Our null hypothesis is that these two samples are from populations with equal means. Alternative hypothesis: population mean for incongruate word order is larger (people need more time to 'parse' it). Use `t.test` function to perform a test. Don't forget to specify `alternative`.

Would you reject null hypothesis (under 5% significance level) according to this test?

What claim about logarithms of reaction time for Congruent and Incongruent stimuli can you make according to this test?

2.4. Paired t-test: manually

To use paired t-test, let us find difference between logarithms of reaction time for each participant. Use `mutate` and add variable `diff` with aforementioned meaning to dataframe `mean_rtlog`. Save result as `mean_rtlog` again. Then compare mean of `diff` with 0 using 1-sample t-test. (Use appropriate alternative.)

Would you reject null hypothesis?

What claim about logarithms of reaction time for Congruent and Incongruent stimuli can you make now?

How can you interpret difference with the result of 2.3?

2.5. Paired t-test out of the box

In fact, we can avoid manual calculation of difference and perform paired t-test using `t.test` function with parameter `paired = True`. Apply this function to your data and make sure you get the same result as in 2.4.

3. Difference between conditions

Now we will consider reaction time for Incongruent word ordering only. Let us check are there any statistically significant difference in logarithm of reaction time for different conditions (types of stimuli).

3.1 Data preparation

Filter only observation with `Incongruent` word order, then find average logarithm of reaction time for each participant and each type of stimuli. Save new dataframe as `incong_rtlog` variable. It should look like the following table:

```
# A tibble: 420 x 3
  ParticipantID      TypeOfStimuli      RTlog
  <fct>           <fct>           <dbl>
1 00e0b159cf5b9abcc73b92506d8b1c38 Shape-Color      3.34
2 00e0b159cf5b9abcc73b92506d8b1c38 Size-Nationality 3.20
3 00e0b159cf5b9abcc73b92506d8b1c38 SubjectiveComment-Material 3.19
4 021a49cde484f8fa18439f026ec99459 Shape-Color      3.20
```

3.2 Statistical testing

Use appropriate statistical test to answer the following question: are there any statistically significant difference in logarithm of reaction time for different conditions (types of stimuli)? Choose the test and provide justification for your choice. Provide your code, results and interpretation. What is your answer to the question?

3.3 Post-hoc analysis: which differences are significant?

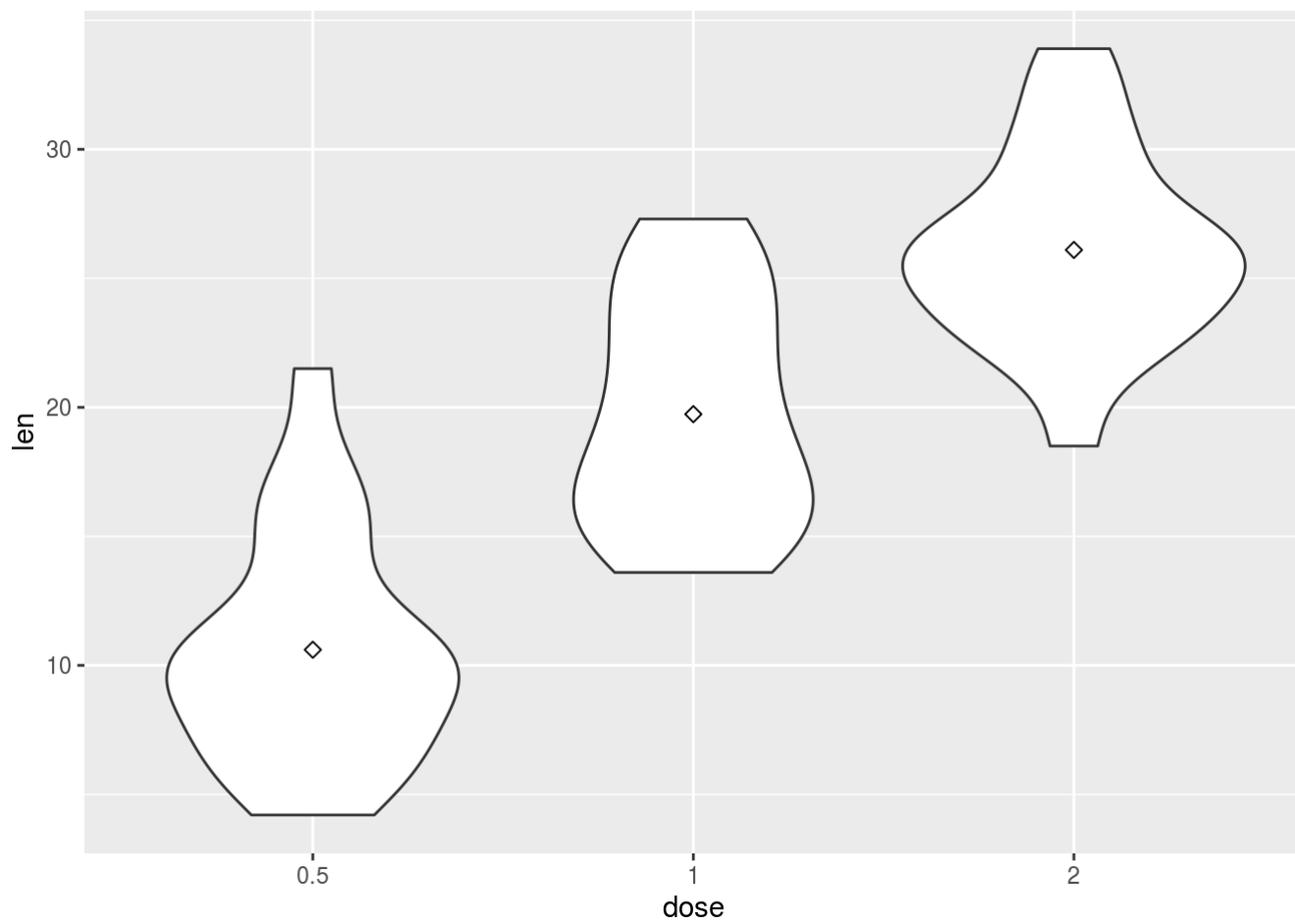
If we compare means for several (more than two) groups and reject null hypothesis that corresponding population means are equal to each other, the next natural question is to find all pairs of groups which difference is statistically significant. As we discussed at the lecture, pairwise t-tests cannot be used here without appropriate corrections. Instead, one can use Tukey Honest Significant Differences. It reports adjusted confidence intervals for differences between group means for each pair of groups as well as p-values for null hypothesis 'difference is equal to zero'.

Apply `TukeyHSD` to the result of 3.2 and report which pair of conditions has statistically significant difference between logarithms of reaction time.

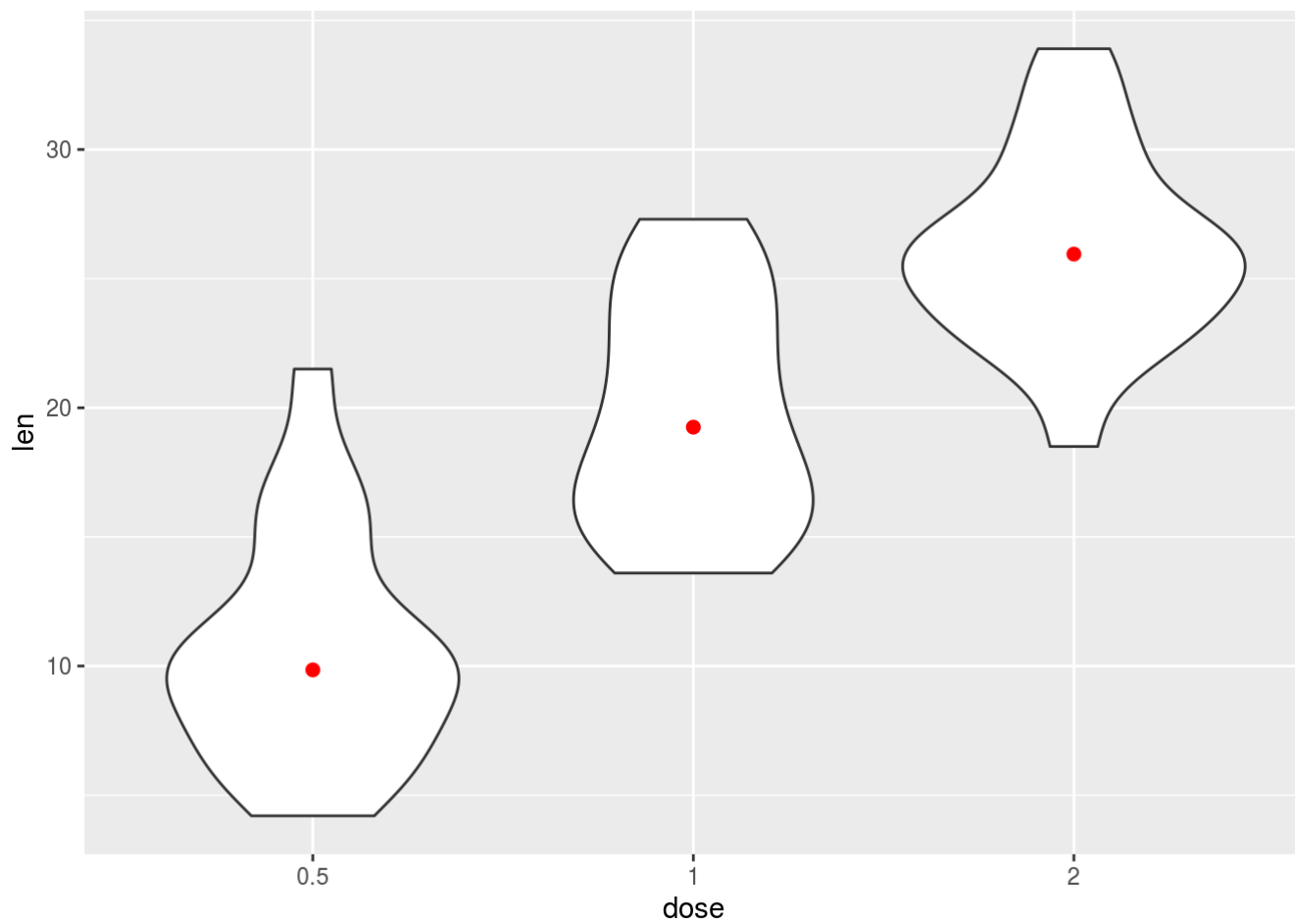
R code cookbook

Violin plot – is similar to box plots, except that they also show the approximation of probability density of the data at different values. Typically, violin plots will include a marker for the median of the data and a box indicating the interquartile range, as in standard box plots.

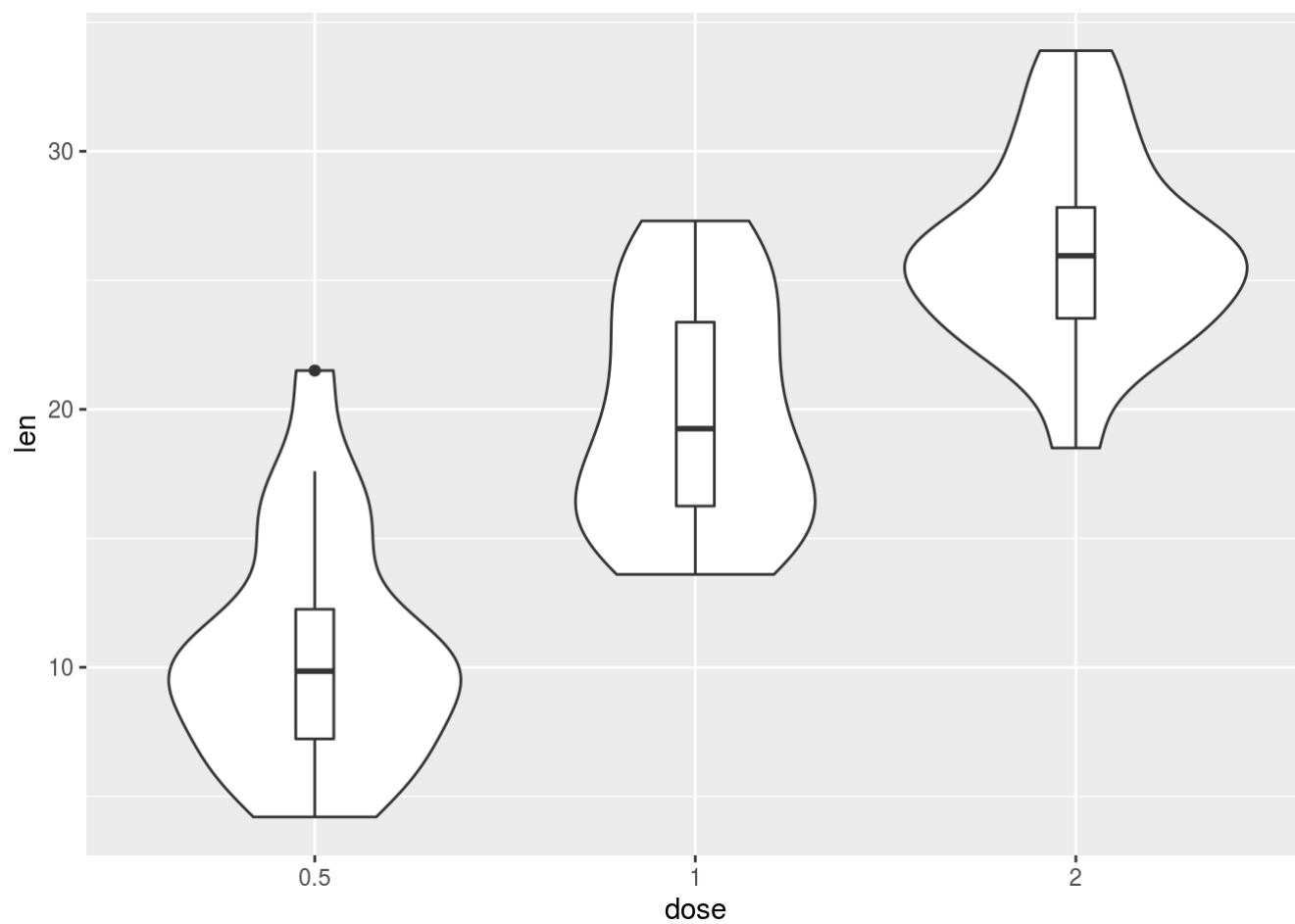
```
ToothGrowth$dose <- as.factor(ToothGrowth$dose)
p <- ggplot(ToothGrowth, aes(x=dose, y=len)) +
  geom_violin()
p + stat_summary(fun.y=mean, geom="point", shape=23, size=2)
```



```
p + stat_summary(fun.y=median, geom="point", size=2, color="red")
```

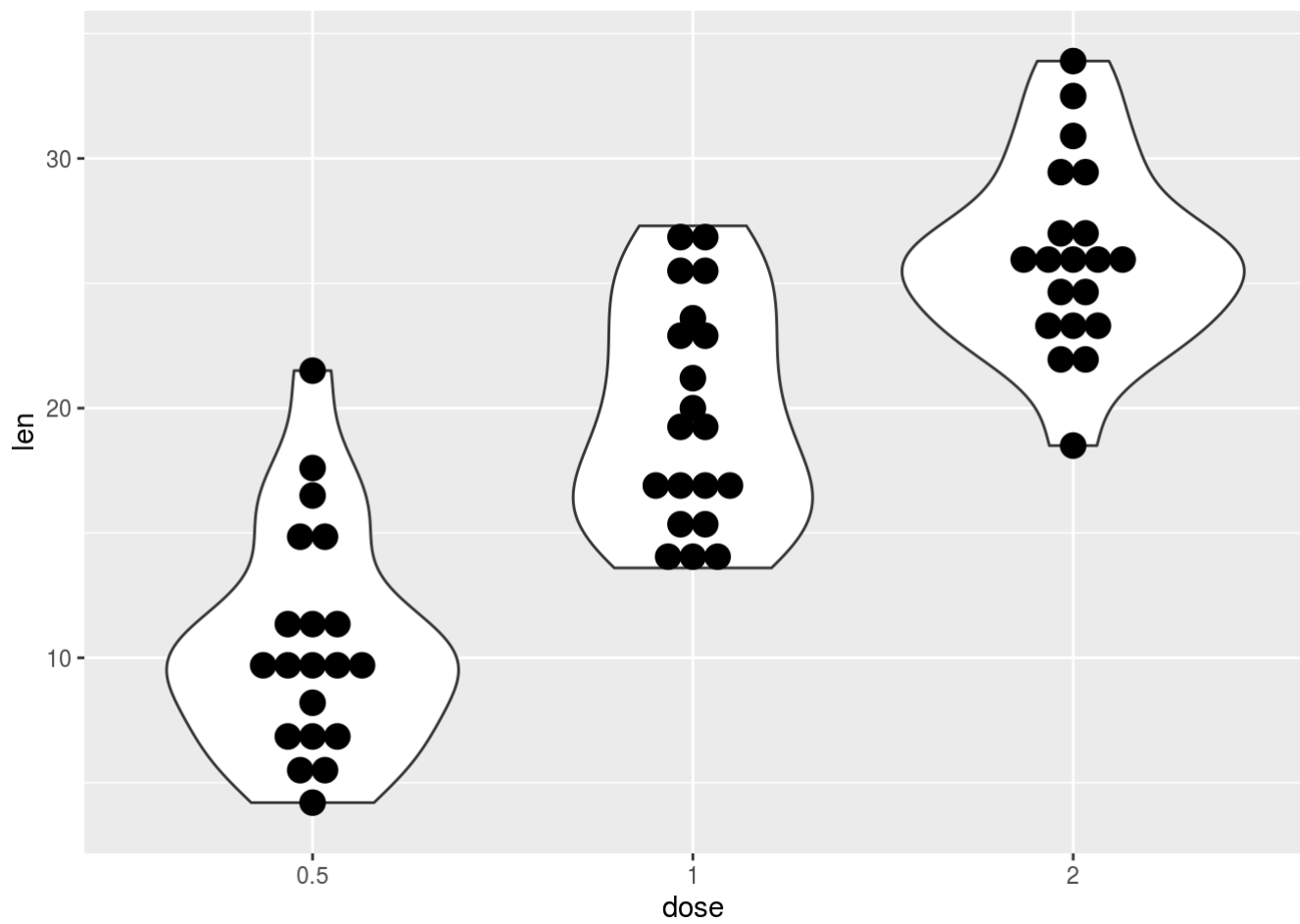


```
p + geom_boxplot(width=0.1)
```

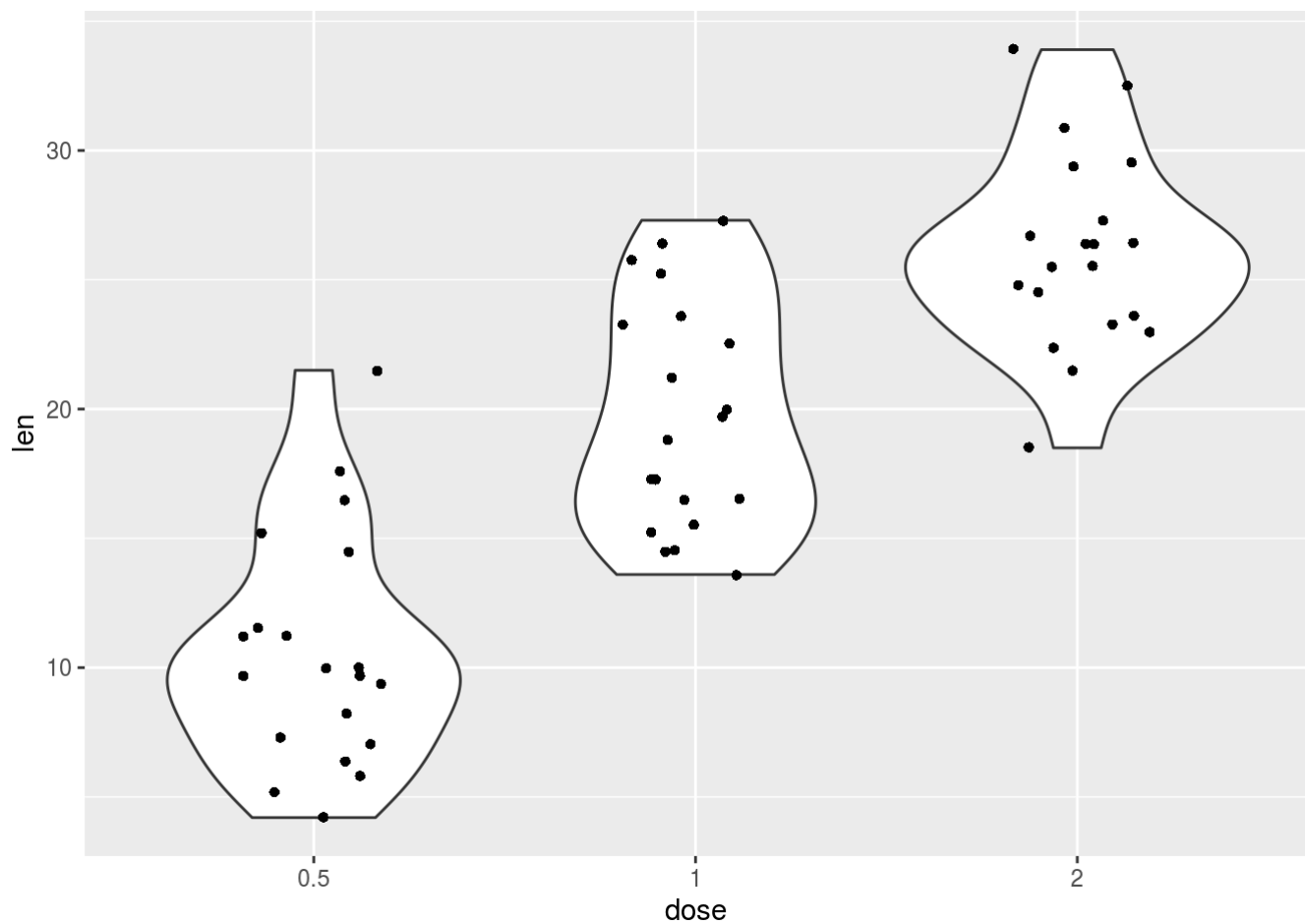


```
# violin plot with dot plot  
p + geom_dotplot(binaxis='y', stackdir='center', dotsize=1)
```

```
## `stat_bindot()` using `bins = 30`. Pick better value with `binwidth`.
```

```
# violin plot with jittered points  
# 0.2 : degree of jitter in x direction  
p + geom_jitter(shape=16, position=position_jitter(0.2))
```



Line plot You can create a line plot of mean \pm error using the function `ggline()` [in `ggpubr`]. Basic line plots of means \pm se with jittered points:

```
library(ggpubr)
```

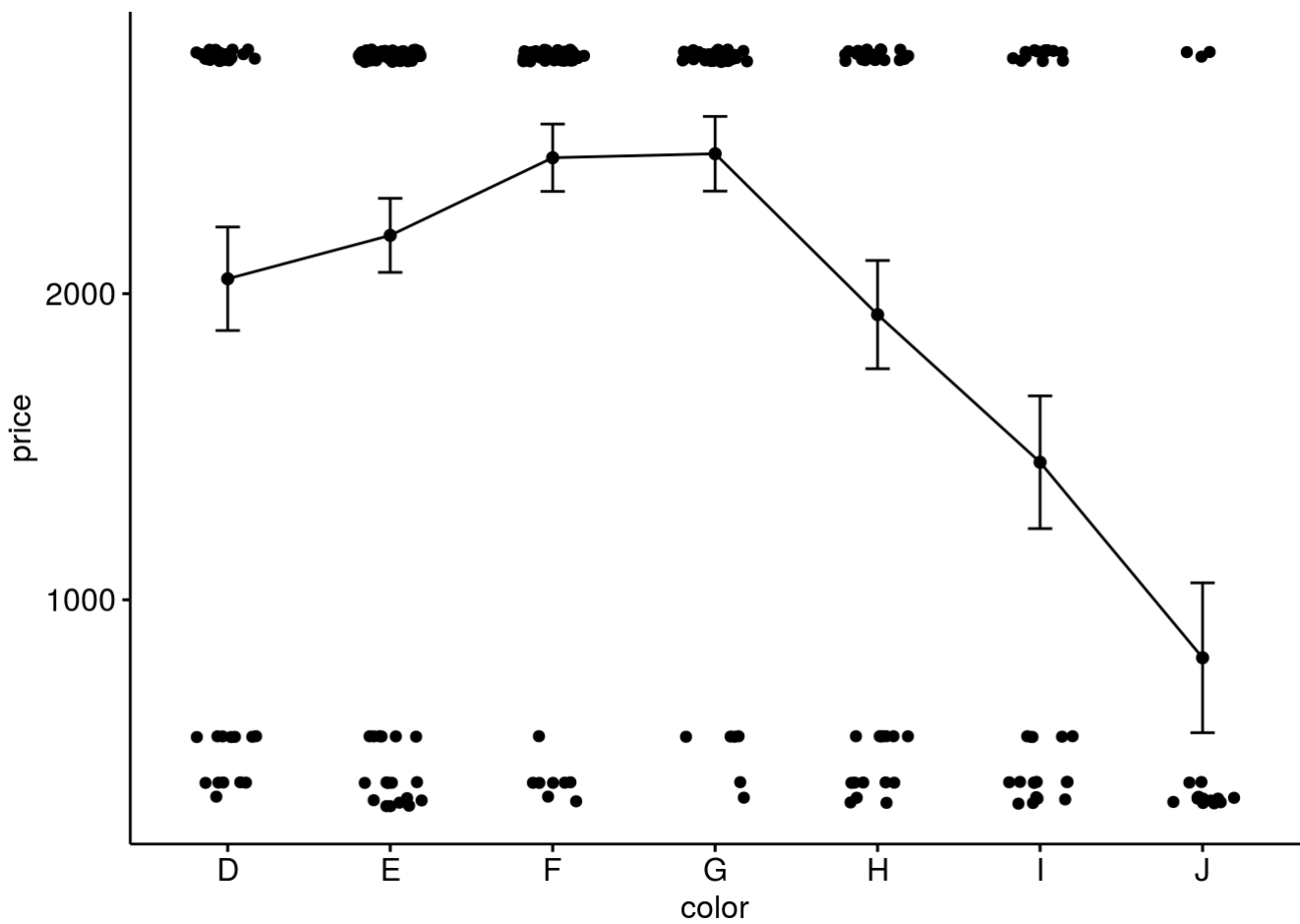
```
## Loading required package: magrittr
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
##
##   set_names
```

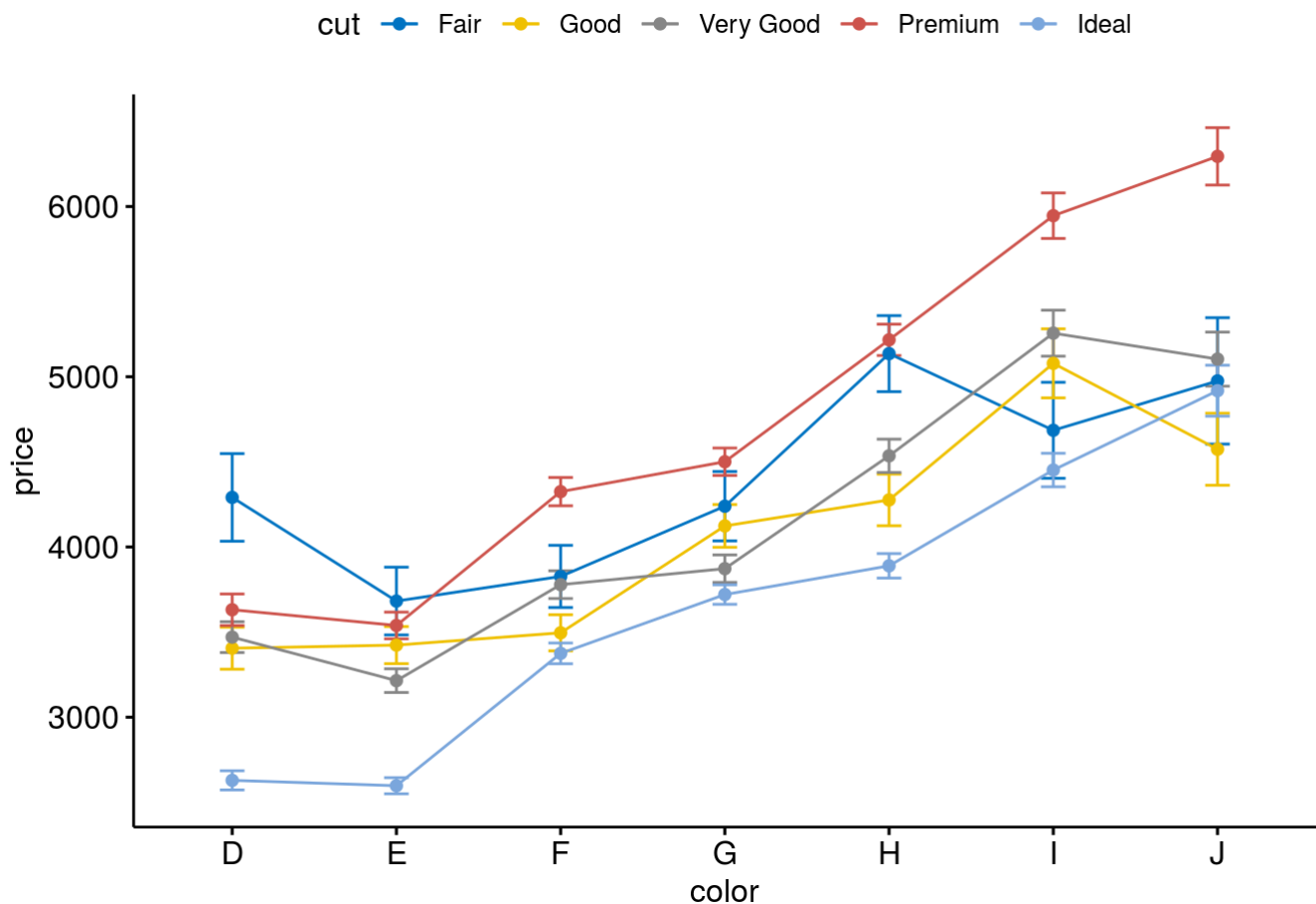
```
## The following object is masked from 'package:tidyr':
##
##   extract
```

```
ggline(diamonds[1:300,], x = "color", y = "price",
       add = c("mean_se", "jitter"))
```



Split and color by group:

```
ggline(diamonds, x = "color", y = "price",  
  add = "mean_se",  
  color = "cut", palette = "jco")
```

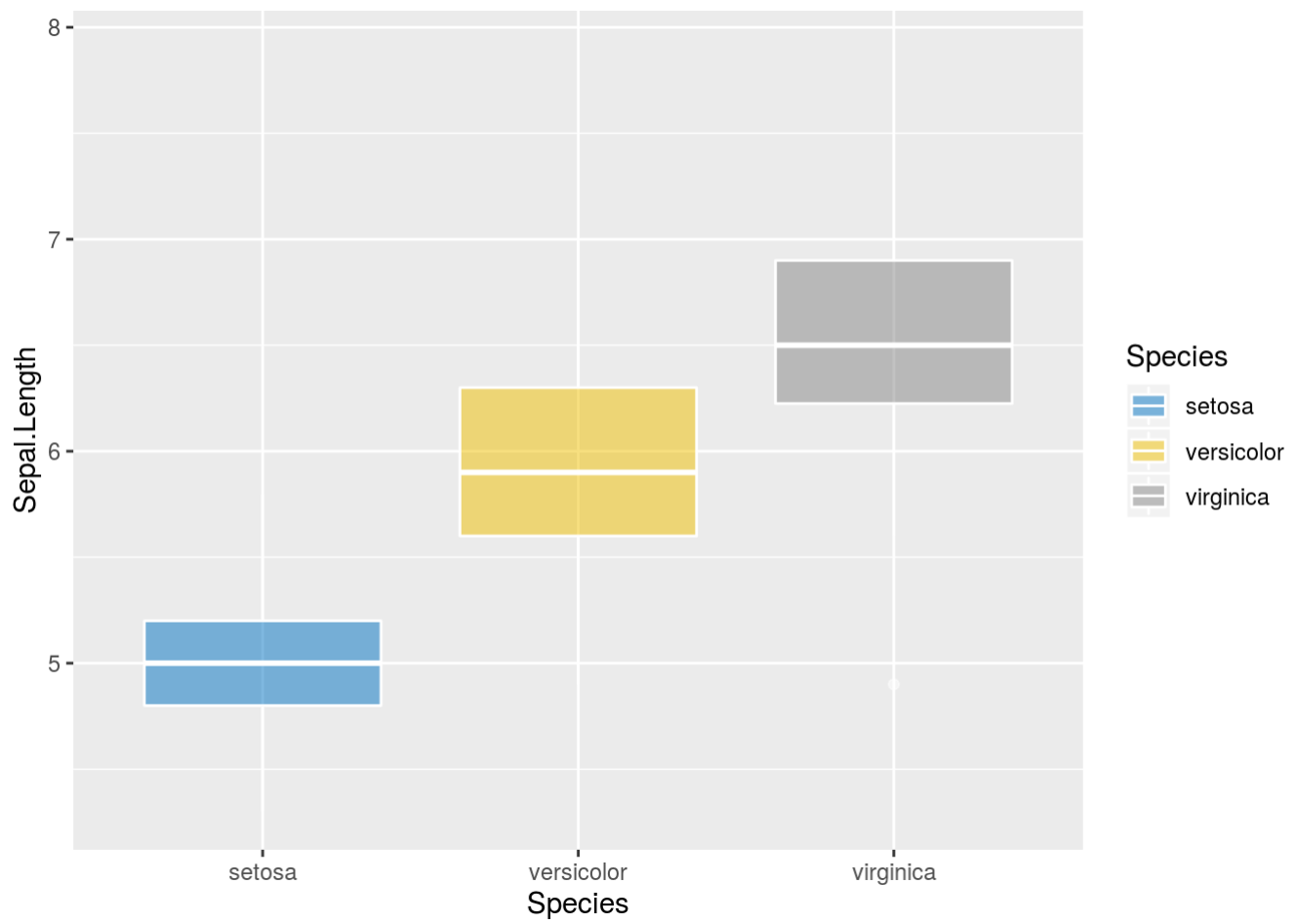


diamonds

```
## # A tibble: 53,940 x 10
##   carat cut      color clarity depth table price     x     y     z
##   <dbl> <ord>    <ord> <ord>  <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1 0.23 Ideal     E    SI2    61.5   55   326  3.95  3.98  2.43
## 2 0.21 Premium  E    SI1    59.8   61   326  3.89  3.84  2.31
## 3 0.23 Good     E    VS1    56.9   65   327  4.05  4.07  2.31
## 4 0.290 Premium I    VS2    62.4   58   334  4.2   4.23  2.63
## 5 0.31 Good     J    SI2    63.3   58   335  4.34  4.35  2.75
## 6 0.24 Very Good J    VVS2    62.8   57   336  3.94  3.96  2.48
## 7 0.24 Very Good I    VVS1    62.3   57   336  3.95  3.98  2.47
## 8 0.26 Very Good H    SI1    61.9   55   337  4.07  4.11  2.53
## 9 0.22 Fair     E    VS2    65.1   61   337  3.87  3.78  2.49
## 10 0.23 Very Good H    VS1    59.4   61   338  4     4.05  2.39
## # ... with 53,930 more rows
```

Using fill_palette

```
ggplot(iris, aes(Species, Sepal.Length))+
  geom_boxplot(aes(fill = Species), color = "white", alpha = 0.5)+
  fill_palette("jco")
```



See more examples of ggpubr plots: here (<https://rpkgs.datanovia.com/ggpubr/index.html>).