factor_example

Why factors are cool!

First we will create some data about absences for different students. Each row is a different student. We have information about the number of days absent and the grade for the individual students. We will use the `tibble()` function to create the data. We will use the `sample()` function to create a random sequence of numbers from 0 to 7 with replacements for 32 hypothetical students. Since there are four grades and 8*4 is 32, we will repeat the grade values 8 times. We use the `set.seed()` function so that the random sample from 0 to 7 is the same each time the code is run.

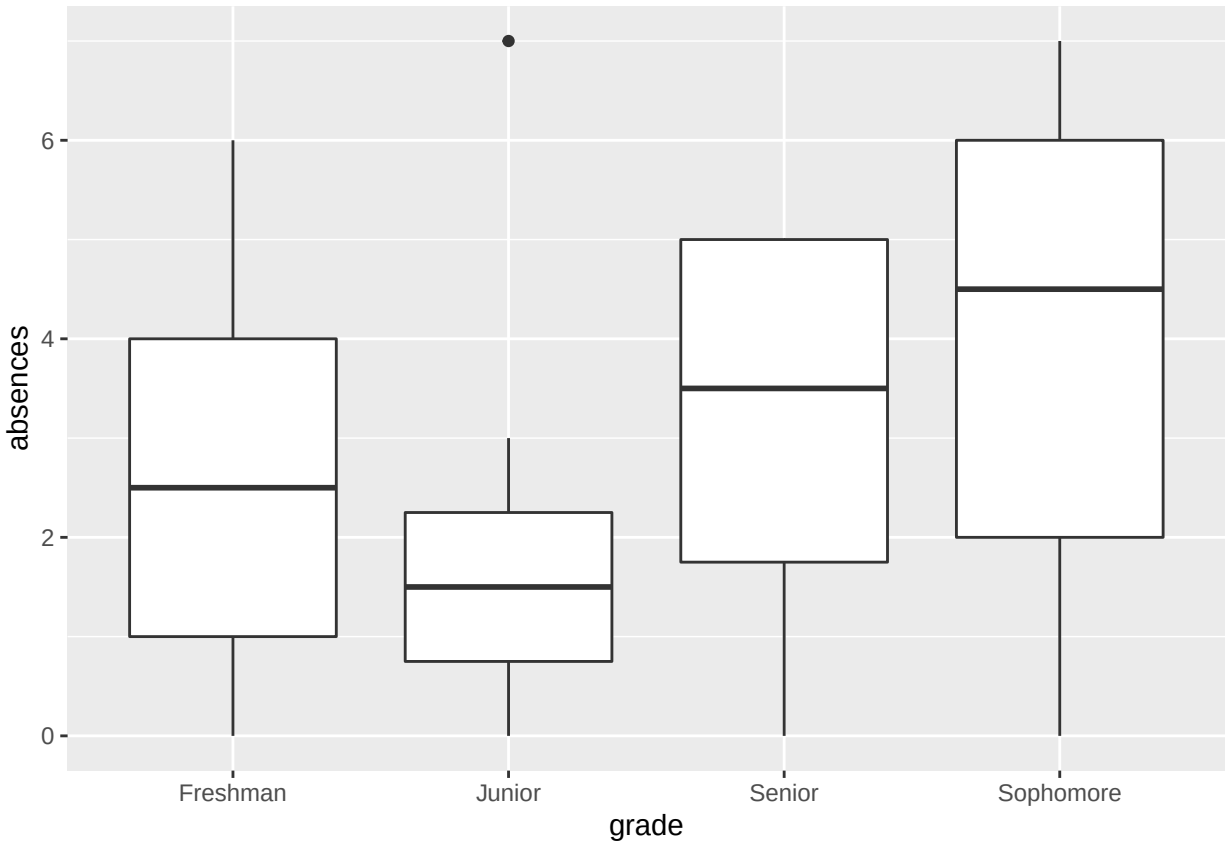
```
set.seed(123)
data_highschool <- tibble(absences = sample(0:7, size = 32, replace = TRUE), grade = rep(c("Sophomore",
data_highschool
```

```
## # A tibble: 32 x 2
##    absences grade
##       <int> <chr>
##  1        6 Sophomore
##  2        6 Freshman
##  3        2 Junior
##  4        5 Senior
##  5        2 Sophomore
##  6        1 Freshman
##  7        1 Junior
##  8        5 Senior
##  9        2 Sophomore
## 10        4 Freshman
## # ... with 22 more rows
```

Notice that `grade` is a `chr` variable. This indicates that the values are character strings. R does not realize that there is any order related to the `grade` values. However, we know that the order is: freshman, sophomore, junior, senior.

Let's make a plot first:

```
data_highschool %>%
  ggplot(mapping = aes(x = grade, y = absences)) +
  geom_boxplot()
```



OK this is very useful, but it is a bit difficult to read, because we expect the values to be plotted by the order that we know, not by alphabetical order. Currently `grade` is class `character` but let's change that to class `factor` which allows us to specify the levels or order of the values.

```
class(data_highschool$grade)
```

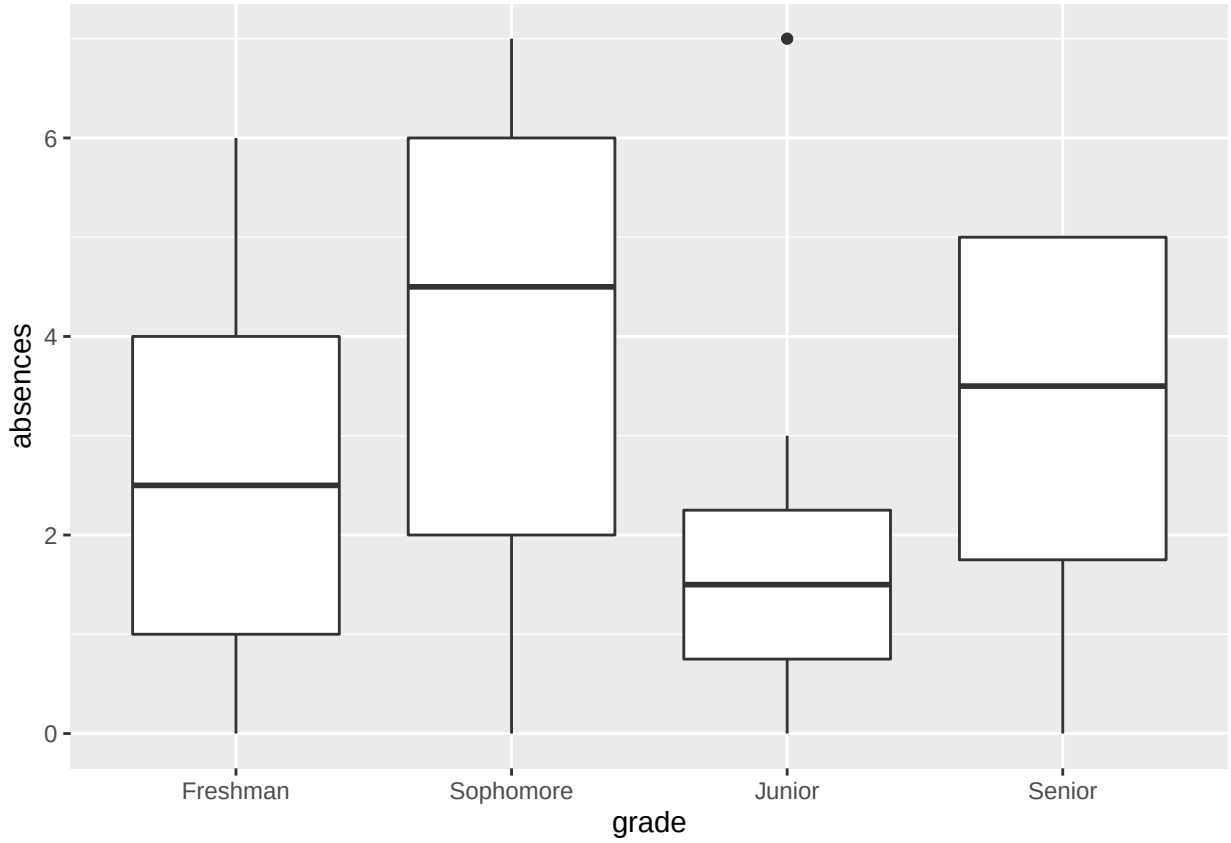
```
## [1] "character"
```

```
data_highschool_fct <- data_highschool %>%
  mutate(grade = factor(grade, levels = c("Freshman", "Sophomore", "Junior", "Senior")))
data_highschool_fct
```

```
## # A tibble: 32 x 2
##    absences grade
##       <int> <fct>
##  1        6 Sophomore
##  2        6 Freshman
##  3        2 Junior
##  4        5 Senior
##  5        2 Sophomore
##  6        1 Freshman
##  7        1 Junior
##  8        5 Senior
##  9        2 Sophomore
## 10        4 Freshman
## # ... with 22 more rows
```

Now let's make our plot again:

```
data_highschool_fct %>%
  ggplot(mapping = aes(x = grade, y = absences)) +
  geom_boxplot()
```



Now that's more like it! Notice how the data is automatically plotted in the order we would like.

What about if we arrange the two versions of our data by grade?

```
data_highschool %>% arrange(grade)
```

```
## # A tibble: 32 x 2
##    absences grade
##       <int> <chr>
##  1        6 Freshman
##  2        1 Freshman
##  3        4 Freshman
##  4        0 Freshman
##  5        4 Freshman
##  6        3 Freshman
##  7        2 Freshman
##  8        1 Freshman
##  9        2 Junior
## 10        1 Junior
## # ... with 22 more rows
```

```
data_highschool_fct %>% arrange(grade)
```

```
## # A tibble: 32 x 2
##    absences grade
##       <int> <fct>
##  1        6 Freshman
##  2        1 Freshman
##  3        4 Freshman
##  4        0 Freshman
##  5        4 Freshman
##  6        3 Freshman
##  7        2 Freshman
##  8        1 Freshman
##  9        6 Sophomore
## 10        2 Sophomore
## # ... with 22 more rows
```

Again notice that the order is not what we would hope for with the first version, but it is for the second version!

Now what about results from some calculations.

```
data_highschool %>% group_by(grade) %>% summarise(mean = mean(absences))
```

```
## # A tibble: 4 x 2
##   grade      mean
##   <chr>     <dbl>
## 1 Freshman   2.62
## 2 Junior     2
## 3 Senior     3.12
## 4 Sophomore  4
```

```
data_highschool_fct %>% group_by(grade) %>% summarise(mean = mean(absences))
```

```
## # A tibble: 4 x 2
##   grade      mean
##   <fct>     <dbl>
## 1 Freshman   2.62
## 2 Sophomore  4
## 3 Junior     2
## 4 Senior     3.12
```

Here we see that the mean is calculated in the order we would like only for the version of the data that has absences coded as a factor!