STAT 145 Examples, Unit 1 Day 4

T.Scofield

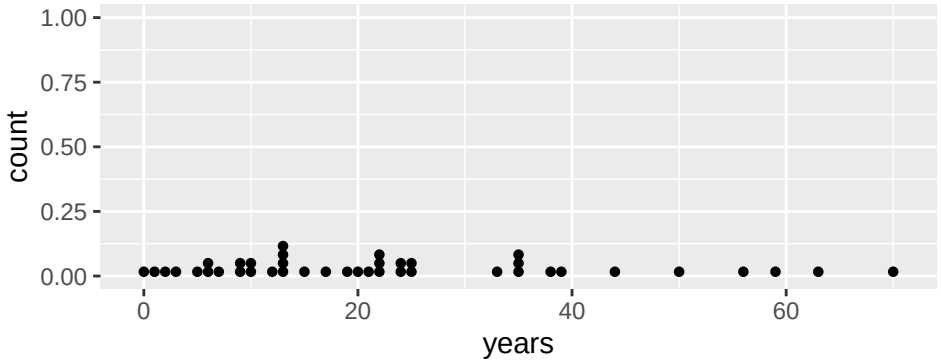
Quantiles

I'll load the English monarchs data set and look at a dot plot of the `years` variable:

```
englishMonarchs <- read.csv("https://scofield.site/teaching/data/csv/monarchReigns.csv")
nrow(englishMonarchs)    # counts the number of cases
```

```
[1] 41
```

```
gf_dotplot(~years, data=englishMonarchs, dotsize=0.8, binwidth=1)
```



There are 41 dots, corresponding to the reigns of the 41 monarchs in the data frame. If you start from the left and count to the 21st dot, that is the **median**, the 50th percentile, with 20 dots on the left and 20 on the right. It's a little more difficult to decide how many dots are to the left and right of the 34th percentile (also known as the 0.34 quantile); commands such as `qdata()` will generally interpolate the position.

```
qdata(~years, data=englishMonarchs, p=.34)
```

```
34%
 13
```

We can ask `qdata()` for multiple percentiles/quantiles at the same time. Here I ask for the 30th, 31st, 32nd, ..., 40th percentiles all at once.

```
qdata(~years, data=englishMonarchs, p=c(.3,.31,.32,.33,.34,.35,.36,.37,.38,.39,.4))
```

```
 30%  31%  32%  33%  34%  35%  36%  37%  38%  39%  40%
12.0 12.4 12.8 13.0 13.0 13.0 13.0 13.0 13.0 13.0 13.0
```

The result is interesting. The command estimated each of the 33rd through 40th percentiles all to be the same, the number 13.0.

The Five Number Summary for quantitative data

If we make no specific request to `qdata()` for which percentiles, we obtain the **five-number summary**.

```
qdata(~years, data=englishMonarchs)
```

```
  0%  25%  50%  75% 100%
   0   10   20   35   70
```

The `fivenum()` command produces the five number summary, as well. When there are only a few numbers to compute these quantiles, be warned that `qdata()` and `fivenum()` do not necessary obtain the same results.

```
smallList = c(3,12,15,16,16,17,19,34)
fivenum(~smallList)
```

```
[1]  3.0 13.5 16.0 18.0 34.0
```

```
qdata(~smallList)      # Note the two commands disagree about 1st/3rd quartiles
```

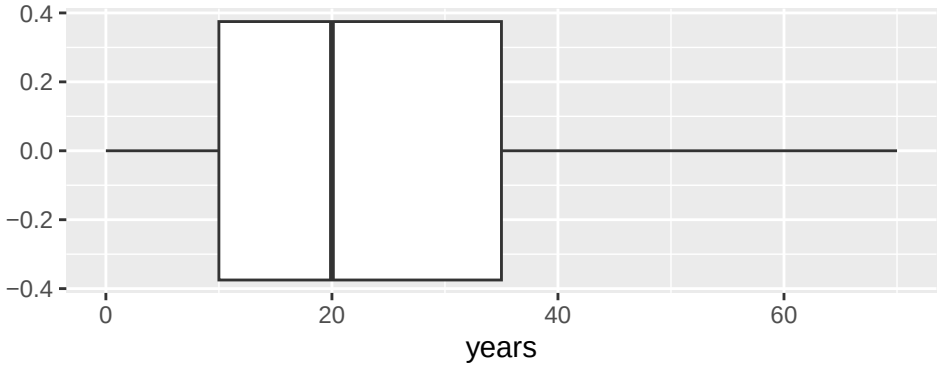
```
   0%   25%   50%   75%  100%
 3.00 14.25 16.00 17.50 34.00
```

- simple random sample (SRS)
- independent and identically distributed (iid) sample

Box-and-Whisker plots

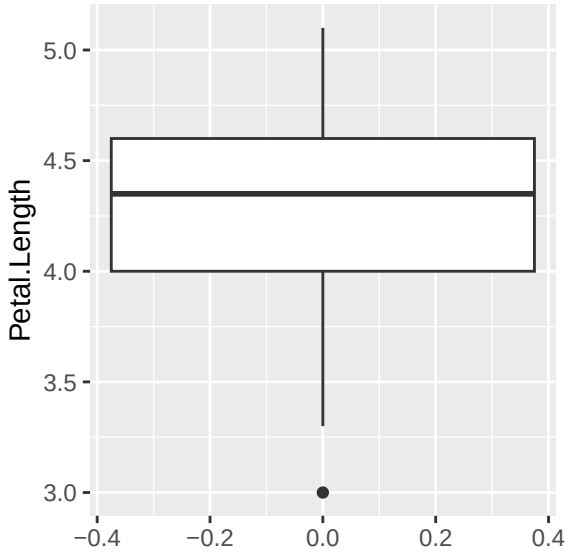
Box-and-whisker plots, also known as **boxplots**, can take arguments similar to those of other commands. Here is one for the reigns of English monarchs:

```
gf_boxplot(~years, data=englishMonarchs)
```



To obtain a boxplot of `Petal.Length` for only those plants in `iris` from the "versicolor" `Species`, we can filter down to the "versicolor" cases:

```
gf_boxplot(Petal.Length ~ ., data=filter(iris, Species=="versicolor"))
```



I have changed the syntax somewhat in order to orient the box vertically instead of horizontally. Notice that a focus on these specific plants results in one point flagged as an **outlier** based on the 1.5×IQR-rule. Note, too, that while one axis is meaningful, the other is not. Do you think it just as easy to estimate the IQR from a boxplot as it is from the 5-number summary?

Variance and standard deviation

In class we computed the variance and standard deviation of a set of four numbers, a very small quantitative data set. Of course, R is capable of carrying out the computation.

```
smallList <- c(11, 16, 38, 23)
var(~smallList)     # command to calculate variance
```

```
[1] 138
```

```
sd(~smallList)      # command to calculate standard deviation
```

```
[1] 11.74734
```

As you would expect, $\sqrt{138} \doteq 11.747$. Hand calculations would be no different on a large data set, only more tedious. Here we use formula notation to compute the standard deviations for `Petal.Length` for the 50 iris plants of each of the three `Species`:

```
sd(~Petal.Length | Species, data=iris)
```

```
    setosa versicolor  virginica
 0.1736640  0.4699110  0.5518947
```

Some Further Questions (for practice)

Q1: How might you find the range of `Petal.Length` for these 3 species?

Answer 1: One possibility is to find the minimum (or 0th percentile) and maximum (or 100th percentile) for `Petal.Length`, broken down by `Species`:

```
qdata(~ Petal.Length | Species, data=iris, p=c(0,1))
```

```
     Species  0% 100%
1     setosa 1.0  1.9
2 versicolor 3.0  5.1
3  virginica 4.5  6.9
```

Subtracting gives us the result.

Q2: How might you find the IQR of `Petal.Length` for these 3 species?

Answer 2: This time let's find the full five-number summaries broken down by `Species`:

```
qdata(~ Petal.Length | Species, data=iris)
```

```
     Species  0% 25%  50%   75% 100%
1     setosa 1.0 1.4 1.50 1.575  1.9
2 versicolor 3.0 4.0 4.35 4.600  5.1
3  virginica 4.5 5.1 5.55 5.875  6.9
```

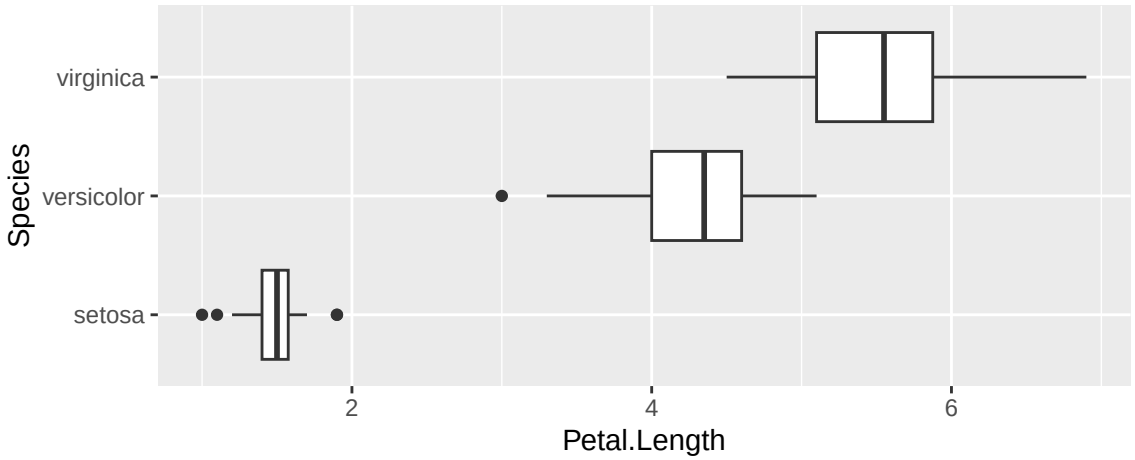
Instead of subtracting minima from maxima, we obtain IQRs by subtracting 1st quartiles from 3rd quartiles. The species with the largest IQR is "virginica", with an IQR $= 5.875 - 5.1 = 0.775$.

Q3: Can we obtain side-by-side boxplots broken down by species?

Answer 3: Indeed, it is possible. Here are two versions. One produces horizontally-oriented boxplots, while the other produces vertical ones. I've only allowed the software to display the latter.

```
gf_boxplot(Petal.Length ~ Species, data=iris)
```

```
gf_boxplot(Species ~ Petal.Length, data=iris)
```


virginica
versicolor
setosa
Species
2
4
6
Petal.Length