DapR1: Notes on the Live R Session, Week 7

In this week's Live R, we'll be reviewing some functions and operators that allow us to check for specific values within our data and subset or filter our data according to specific constraints. We'll also learn some ways to visualize our data using tables.

First, let's load the tidyverse.

```
library(tidyverse)
library(ggmosaic)
```

Today we're going to be using different types of colourful candy to illustrate sets, sample spaces, and probability.

Set Operations

Let's create some data to compare. We have two subsets of candy colours corresponding to the basic colours of M&Ms and Skittles.

```
mms <- c('brown', 'yellow', 'green', 'red', 'orange', 'blue')
skittles <- c('red', 'green', 'yellow', 'purple', 'orange')
```

We can show which values are elements of both sets by using the *intersect()* function:

```
intersect(mms, skittles)
```

```
## [1] "yellow" "green"  "red"    "orange"
```

We can see which values are elements of only one of the two sets by using the *setdiff()* function. Note that it's not symmetrical, and outputs the values in x that are not in y.

```
setdiff(mms, skittles)
```

```
## [1] "brown" "blue"
```

```
setdiff(skittles, mms)
```

```
## [1] "purple"
```

We can also use the *setequal()* function to tell us if our two sets are equal.

```
setequal(mms, skittles)
```

```
## [1] FALSE
```

```
x <- 1:5
y <- 5:1
setequal(x, y)
```

```
## [1] TRUE
```

Sample Space

Let's imagine we are interested in the likelihood of choosing each colour from a bag of Skittles. In each bag, there are 5 colour choices. These will be a part of our **sample space**, which is the collection of all possible outcomes. If we were to select one skittle, our sample space is each colour of skittle:

```
skittles
```

```
## [1] "red"    "green"  "yellow" "purple" "orange"
```

According to Mars Wrigley, there should be an equal proportion of each colour in a bag. In other words, if you randomly select a single Skittle from a bag, you have a 20% chance of getting any of the 5 colours. Imagine we have an magic bag that produces an unlimited number of Skittles (so that the probability of choosing each colour isn't affected by the total number of that colour selected already). Let's see how the frequency with which we observe each colour fluctuates as number of samples increases:

```
set.seed(2210)
s5 <- tibble(Obs = sample(skittles, 5, replace = T))
s100 <- tibble(Obs = sample(skittles, 100, replace = T))
s5
```

```
## # A tibble: 5 x 1
##   Obs
##   <chr>
## 1 red
## 2 yellow
## 3 yellow
## 4 yellow
## 5 yellow
```

We can look at the frequency of each colour using tables. You could look at raw counts or the proportion of each colour.

```
# Raw Frequencies
s5 %>%
  table()
```

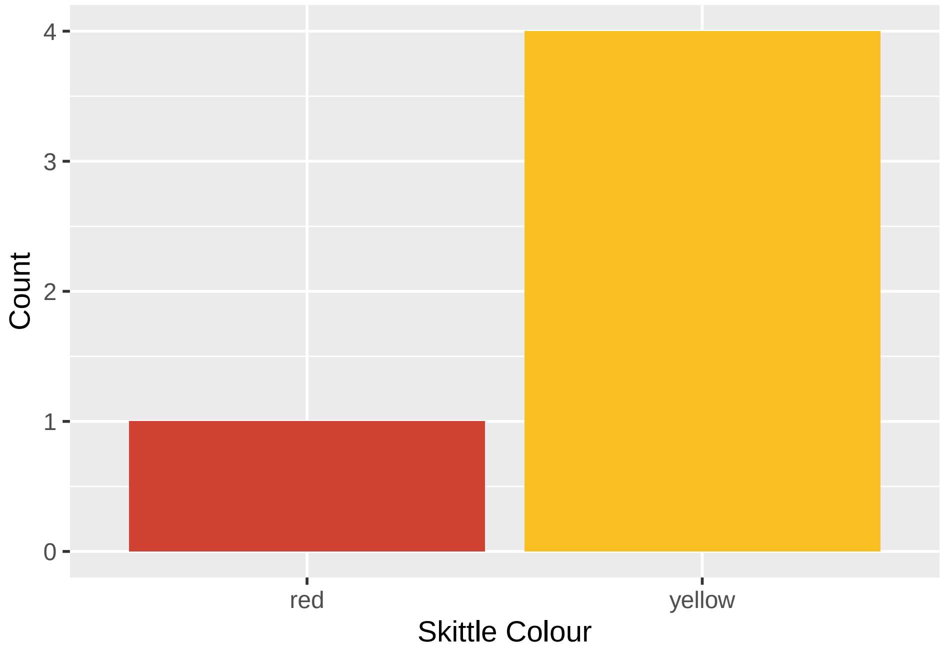
```
## .
##    red yellow
##      1      4
```

```
# Proportion
s100 %>%
  table %>%
  prop.table()
```

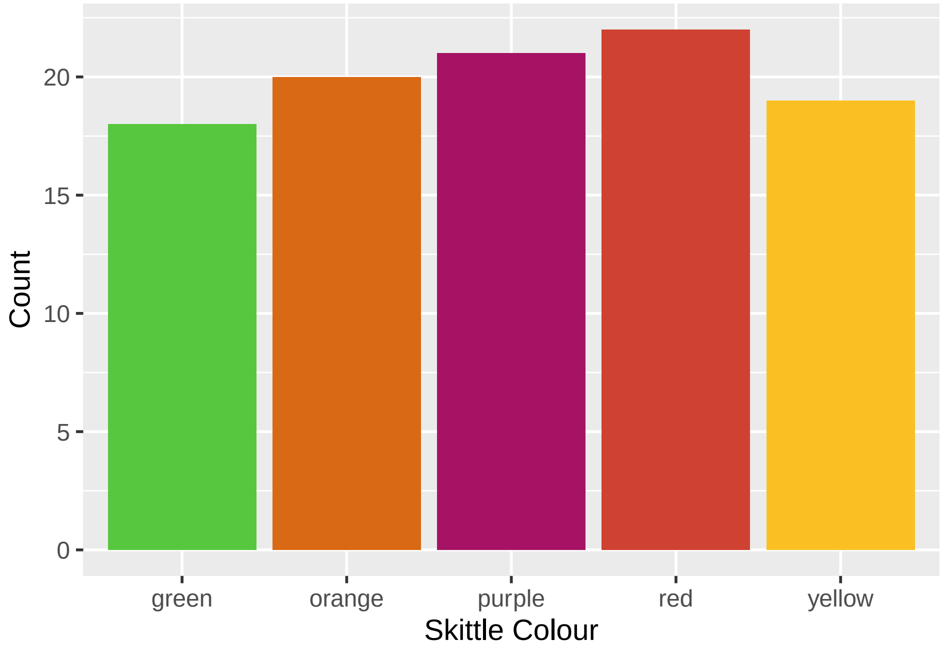
```
## .
##  green orange purple    red yellow
##   0.18   0.20   0.21   0.22   0.19
```

We can also visualize the results using frequency plots, such as bar plots and histograms. In this case, we would use bar plots because the value on our x-axis (*Skittle Colour*) is categorical.

```
ggplot(s5, aes(Obs, fill = Obs)) + geom_bar() +
  labs(x = 'Skittle Colour', y = 'Count') +
  scale_fill_manual(values = c('#CF4232', '#FAC023')) +
  theme(legend.position = 'none')
```



```
s100 %>%
  ggplot(aes(Obs, fill = Obs)) + geom_bar() +
  labs(x = 'Skittle Colour', y = 'Count') +
  scale_fill_manual(values = c('#57C73F', '#D96915', '#A61364', '#CF4232', '#FAC023')) +
  theme(legend.position = 'none')
```



You'll notice that the proportions of each colour in the larger sample will look more like what we would expect (e.g., ~20% for each). This links to the **law of large numbers** discussed in the lecture and lab.

Events

An **event** is a subset of all possible outcomes. Say we were interested only in the probability of selecting a green skittle from the bag. In this case, the event would be:

```
grEvent <- 'green'
```

If we select a skittle from the bag, we can see whether the outcome is consistent with our event of interest using the == operator. This operator can be used to determine whether a given value is exactly equal to another value. Note that it should not be confused with the = operator, which is used for assignment.

```
draw1 <- sample(skittles, 1)
draw1
```

```
## [1] "orange"
```

```
draw1 == grEvent
```

```
## [1] FALSE
```

Remember from the lecture that the probability equation is

$P(x) = \frac{total\ successes}{total\ successes + total\ fails}$

We can use the == operator to compute the total number of successful green skittle selections in our small and large samples. We will use the opposite of the == operator, **!=**, to compute the number of failures to select a green skittle. The != operator allows us to specify all values *not equal to* a specific value.

```r
# total green picks/(total green picks + total non-green picks)
sum(s5 == grEvent)/(sum(s5 == grEvent) + sum(s5 != grEvent))
```

```
## [1] 0
```

```r
sum(s100 == grEvent)/(sum(s100 == grEvent) + sum(s100 != grEvent))
```

```
## [1] 0.18
```

Note that the == and != operators require an exact match and are case sensitive, so they work best when you're working with a single specific value.

```r
s5 == 'yellow'
```

```
##        Obs
## [1,] FALSE
## [2,]  TRUE
## [3,]  TRUE
## [4,]  TRUE
## [5,]  TRUE
```

```r
s5 == 'Yellow'
```

```
##        Obs
## [1,] FALSE
## [2,] FALSE
## [3,] FALSE
## [4,] FALSE
## [5,] FALSE
```

Joint Probability

With joint probability, we're interested in looking at the probability of multiple outcomes. For example, if we were to select a skittle from the bag, what's the likelihood we would select a yellow **OR** a purple?

Because these events are mutually exclusive, we would expect their joint probability to be the sum of their individual probabilities.

$p(yellow \cup purple) = .20 + .20 = .40$

Here, we'll review two ways you can specify multiple events of interest and use these methods to compute joint probability in your sample.

The AND and OR operators

Two operators that may be used for specifying multiple conditions are **&** and |. As you might expect, **&** refers to *'and'*. It allows you to specify values that meet *all* given conditions. The | operator stands for *'or'* and allows you to specify values that meet *any* of the given conditions.

```
successes <- s100 == 'yellow'|s100 == 'purple'
failures <- s100 != 'yellow' & s100 != 'purple'

sum(successes)/(sum(successes) + sum(failures))
```

```
## [1] 0.4
```

%in% or is.element()

You can get the same results using the **%in%** operator. This operator can be used to determine whether a given element (or values from a vector of elements) is found in a dataset or vector. Also, note the use of another operator. The **!** operator stands for *'not'* and allows you to select values that do not meet a specified condition.

```
events <- c('purple', 'yellow')
successes <- s5$Obs %in% events
failures <- !s100$Obs %in% events

sum(successes)/(sum(successes) + sum(failures))
```

```
## [1] 0.0625
```

You can also use the **is.element()** function in the same way. *is.element(x, y)* is identical to *x %in% y*

```
is.element(s5$Obs, events)
```

```
## [1] FALSE  TRUE  TRUE  TRUE  TRUE
```

```
successes <- is.element(s100$Obs, events)
failures <- !is.element(s100$Obs, events)
sum(successes)/(sum(successes) + sum(failures))
```

```
## [1] 0.4
```

Now, let's compute the probability of selecting a yellow **AND** a purple skittle in two selections.

To do this, we would use we use the following formula:

$p(yellow \cap purple) = .20 * .20 = .04$

Let's check to see if our data follow the expected result. To do this, we'll need to add a new column that represents the second skittle selection:

```
s100$Obs2 <- sample(skittles, 100, replace = T)
head(s100)
```

```
## # A tibble: 6 x 2
##   Obs    Obs2
##   <chr>  <chr>
## 1 green  orange
## 2 yellow orange
## 3 yellow purple
## 4 orange purple
## 5 red    purple
## 6 yellow yellow
```

```
successes <- (s100$Obs=='purple'&s100$Obs2=='yellow')|(s100$Obs=='yellow'&s100$Obs2=='purple')
failures <- !successes

sum(successes)/(sum(successes) + sum(failures))
```

```
## [1] 0.05
```

You can also look at the proportion table for multiple events:

```
s100 %>%
  table %>%
  prop.table()
```

```
##          Obs2
## Obs      green orange purple  red yellow
##   green   0.01   0.07   0.00 0.03   0.07
##   orange  0.05   0.02   0.06 0.04   0.03
##   purple  0.06   0.08   0.04 0.02   0.01
##   red     0.06   0.04   0.07 0.05   0.00
##   yellow  0.03   0.06   0.04 0.02   0.04
```

Note that you can sum the appropriate locations in the proportion table to get the corresponding probability.

```
pTab <- s100 %>%
  table() %>%
  prop.table()

pTab['purple', 'yellow'] + pTab['yellow', 'purple']
```

```
## [1] 0.05
```

However, the table outputs in R are not appropriate for a formal report. If you want to include a table in your report, you should use the kbl() function to produce a formal table to display your data:

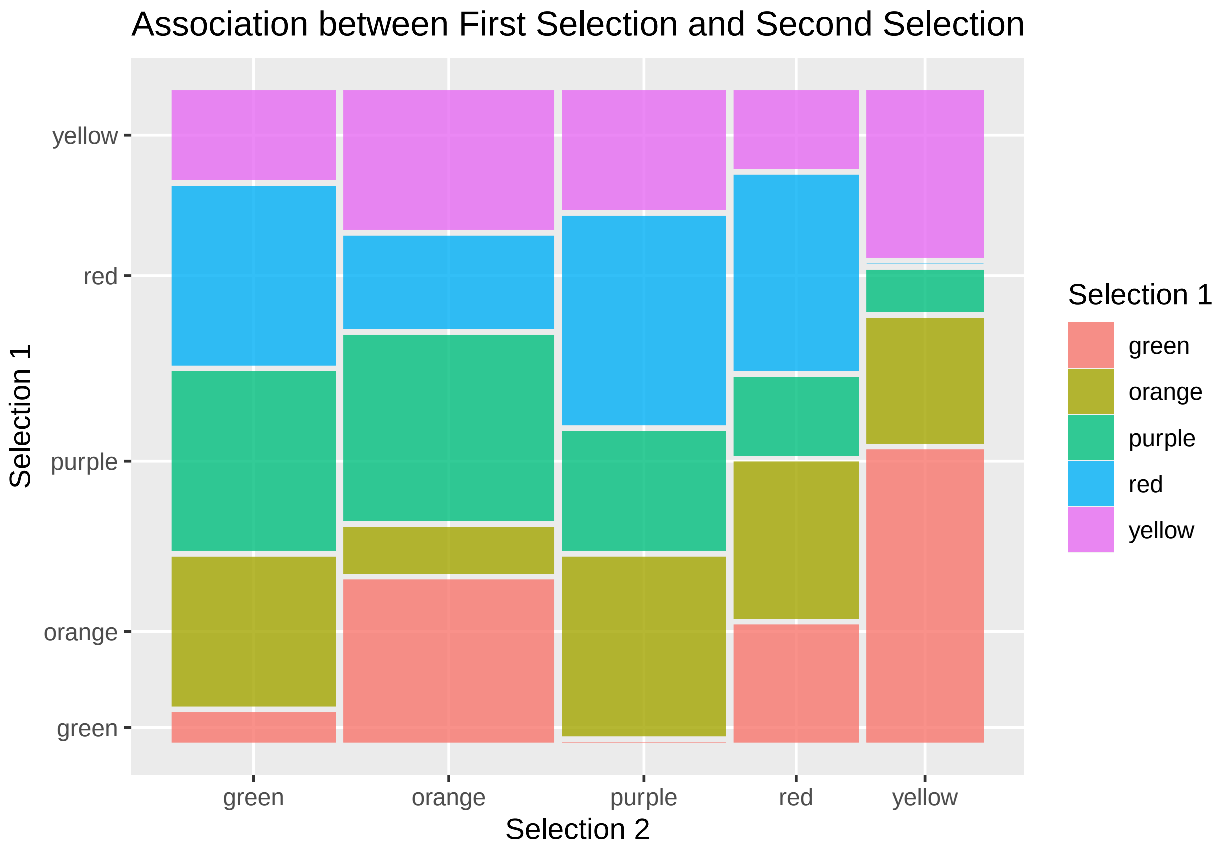
```
library(kableExtra)

pTab %>%
  kbl(booktabs = T, digits = 2, caption = 'Skittles Experiment Data') %>%
  kable_styling(latex_options = c('hold_position', 'striped')) %>%
  column_spec(1, bold = T) %>%
  row_spec(0, bold = T)
```

You can also visualise associations between multiple categorical variables using a mosaic plot.

```
ggplot(s100) +
    geom_mosaic(aes(x = product(Obs, Obs2), fill = Obs)) +
    labs(title = "Association between First Selection and Second Selection",
         x = "Selection 2", y = "Selection 1", fill = 'Selection 1')
```

Table 1: Skittles Experiment Data

	green	**orange**	**purple**	**red**	**yellow**
green	0.01	0.07	0.00	0.03	0.07
orange	0.05	0.02	0.06	0.04	0.03
purple	0.06	0.08	0.04	0.02	0.01
red	0.06	0.04	0.07	0.05	0.00
yellow	0.03	0.06	0.04	0.02	0.04


Association between First Selection and Second Selection
Selection 1
yellow
red
purple
orange
green
green
orange
purple
red
yellow
Selection 2
Selection 1
green
orange
purple
red
yellow