````
---
title: "Introduction to ggplot2"
author: "Derek Sollberger"
date: "August 27, 2017"
output: html_document
---

# Grammar of Graphics

```{r, message = FALSE, warning = FALSE}
library("tidyr")
library("ggplot2")
library("mosaicData")
```

## One-dimensional, continuous data

Let's start with a simple data set where a budding scientist took $n = 30$ dimes and weighed them.

```{r}
data(Dimes)
head(Dimes)
```

We can see that we have data about the mass and the year for each dime.  Treating the years as a *discrete variable*, it may be easist to visualize the years with a `dotplot`.

```{r, message = FALSE, warning = FALSE}
ggplot(Dimes, aes(x = year)) +
  geom_dotplot() +
  ggtitle("Dime Study") +
  ylab("proportion")
```


Treating the masses as a *continuous variable*, it may be best to visualize the masses with a `histogram`.

```{r, message = FALSE, warning = FALSE}
ggplot(Dimes, aes(x = mass)) +
  geom_histogram() +
  ggtitle("Dime Study")
```

We can also change the `binwidth`.

```{r, message = FALSE, warning = FALSE}
ggplot(Dimes, aes(x = mass)) +
  geom_histogram(binwidth = 0.01) +
  ggtitle("Dime Study")
```

---

Let us now look at a more interesting data set.  The `SAT` data features state-by-state average results from SAT tests from 1995.

```{r, message = FALSE, warning = FALSE}
data(SAT)
head(SAT)
```

A *continuous* approximation to the distribution of the data is called a **kernel density estimate**.

```{r, message = FALSE, warning = FALSE}
ggplot(SAT, aes(x = sat)) +
  geom_density(kernel = "gaussian")
```

Where ggplot becomes powerful is the ability to quickly compare different categories.

```{r}
SAT %>%
````
````

```
  gather(key = testType, value = test, verbal, math) %>%
  ggplot(aes(x = test, color = testType)) +
    geom_density(kernel = "gaussian")
```

A couple more aesthetics that we can manipulate include `fill` and `alpha`, where `alpha` is a proportion of how much color to use.

```{r}
SAT %>%
  gather(key = testType, value = test, verbal, math) %>%
  ggplot(aes(x = test, color = testType, fill = testType)) +
    geom_density(kernel = "gaussian", alpha = 0.5)
```

Two, continuous variables

With two continuous variables, we can create a conventional scatterplot with `geom_point()`.

```{r}
ggplot(SAT, aes(x = expend, y = sat)) +
  geom_point()
```

Here we can label the points with another variable.

```{r}
ggplot(SAT, aes(x = expend, y = sat)) +
  geom_point() +
  geom_text(aes(label = state))
```

We can manipulate the placement of the text.

```{r}
ggplot(SAT, aes(x = expend, y = sat)) +
  geom_point() +
  geom_text(aes(label = state), vjust = 2) +
  ggtitle("SAT Scores by State") +
  xlab("education spending") +
  ylab("average SAT score")
```

Facet Grids

Sometimes we want side-by-side graphs.

```{r}
data("SwimRecords")
ggplot(SwimRecords, aes(x = year, y = time)) +
  geom_line() +
  facet_grid(. ~ sex)
```

Color Brewer

```{r}
data("SnowGR")
ggplot(SnowGR, aes(x = Total)) +
  geom_histogram(binwidth = 5)
```

ggplot can utilize colors from `colorBrewer`.

```{r}
ggplot(SnowGR, aes(x = Total, fill = ..x..)) +
  geom_histogram(binwidth = 5) +
  scale_fill_gradient(low = "yellow", high = "blue")
```

Example: Opiate Addiction

```{r}
causes <- c("Poisonings", "Traffic Accidents", "Suicide", "Breast cancer", "Heart Disease")
deaths <- c(6803, 4979, 4159, 2325, 1612)
df <- data.frame(causes, deaths)
ggplot(df, aes(x = causes, y = deaths)) +
  geom_bar(aes(fill = causes), stat = "identity")
```