Reviewing Tables and `ggplot2`

Tyson S. Barrett

Data

For these slides, the activities, and examples we will use the "Office_Parks" data set.

- Contains ficticious data on both *The Office (U.S.)* and *Parks and Recreation* television shows
- Longitudinal (two time points) and nested (within show)
- Currently in wide format

Data can be downloaded from: tysonstanley.github.io

Review of Beautiful Tables

and

`ggplot2` 📊

Load the Data

```
d <- read.csv("OfficeParks.csv")
```

Copy CSV Excel PDF Print Search:

	X	nam	prod1	ment1	phys	marr
1	1	Michael	2	3	8	0
2	2	Pam	3	8	7	1
3	3	Jim	3	8	8	1
4	4	Dwight	5	6	8	0
5	5	Stanley	4	7	4	1

Showing 1 to 5 of 38 entries

Explore the Data

What are some ways to explore this data?

- data summaries via tables
- univariate and multi-variate plots

Quick Summaries

Among the important aspects of the data to explore include:

- correlations
- means and standard deviations
- ranges
- distributions
- missingness

For this we'll use a few functions:

1. `furniture::tableC()`
2. `furniture::table1()`
3. `summary()`
4. `psych::describe()`

Correlations

```
library(furniture)
tableC(d,
       prod1, ment1, depr1, awkw1,
       na.rm=TRUE)
```

```
##
## |=============================================================|
##            [1]             [2]             [3]           [4]
##  [1]prod1 1.00
##  [2]ment1 0.544 (0.001)  1.00
##  [3]depr1 -0.229 (0.207) -0.508 (0.003) 1.00
##  [4]awkw1 -0.365 (0.04)  -0.31 (0.084)  0.05 (0.787) 1.00
## |=============================================================|
```

Descriptives

Simple

```
library(furniture)
table1(d,
       prod1, ment1, depr1, awkw1)
```

```
##
## |==============================|
##                Mean/Count (SD/%)
##  Observations 38
##  prod1
##                3.2 (1.4)
##  ment1
##                5.3 (2.2)
##  depr1
##                11.3 (4.1)
##  awkw1
##                7.6 (5.9)
## |==============================|
```

Descriptives

Stratified

```
library(furniture)
table1(d,
       prod1, ment1, depr1, awkw1,
       splitby = ~show)
```

```
##
## |======================================|
##                       show
##                Parks and Rec The Office
##  Observations 22             16
##  prod1
##                3.1 (1.6)     3.2 (1.1)
##  ment1
##                5.4 (2.3)     5.2 (2.2)
##  depr1
##                10.6 (4.5)    12.1 (3.7)
##  awkw1
##                12.2 (4.6)    2.6 (1.3)
## |======================================|
```

Descriptives

Stratified with bivariate statistical tests (by show)

```
library(furniture)
table1(d,
       prod1, ment1, depr1, awkw1,
       splitby = ~show, test = TRUE)
```

```
##
## |=============================================|
##                            show
##                Parks and Rec The Office P-Value
##  Observations  22              16
##  prod1                                    0.693
##                3.1 (1.6)       3.2 (1.1)
##  ment1                                    0.838
##                5.4 (2.3)       5.2 (2.2)
##  depr1                                    0.314
##                10.6 (4.5)      12.1 (3.7)
##  awkw1                                    <.001
##                12.2 (4.6)      2.6 (1.3)
## |=============================================|
```

Descriptives

```
##
## |=============================================|
##                            show
##                Parks and Rec The Office P-Value
##  Observations  22            16
##  prod1                                   0.693
##                3.1 (1.6)     3.2 (1.1)
##  ment1                                   0.838
##                5.4 (2.3)     5.2 (2.2)
##  depr1                                   0.314
##                10.6 (4.5)    12.1 (3.7)
##  awkw1                                   <.001
##                12.2 (4.6)    2.6 (1.3)
## |=============================================|
```

Any surprises?

- awkw1 seems massively different
- Could be a problem with the data

Descriptives

```
##
## |=============================================|
##                            show
##                Parks and Rec The Office P-Value
##  Observations  22              16
##  prod1                                    0.693
##                3.1 (1.6)       3.2 (1.1)
##  ment1                                    0.838
##                5.4 (2.3)       5.2 (2.2)
##  depr1                                    0.314
##                10.6 (4.5)      12.1 (3.7)
##  awkw1                                    <.001
##                12.2 (4.6)      2.6 (1.3)
## |=============================================|
```

Any surprises?

- `awkw1` seems massively different
- Could be a problem with the data

We can see if there are weird things in the ranges and distributions

Ranges and Distributions

```
library(tidyverse)   ## for the pipe and select()
d %>%
  select(awkw1, awkw2) %>%
  psych::describeBy(group = d$show)
```

```
## Parks and Rec

##       vars  n   mean    sd median trimmed   mad min max range
## awkw1    1 17 12.235 4.562     10  12.333 7.413   4  19    15 ...
## awkw2    2 22 14.909 7.030     13  14.889 9.637   4  26    22 ...

## ------
## The Office

##       vars  n  mean    sd median trimmed   mad min max range
## awkw1    1 16 2.625 1.258      3   2.571 1.483   1   5     4 ...
## awkw2    2 16 1.938 1.769      2   1.786 2.224   0   6     6 ...
```

All descriptives suggest there is a problem with the `awkw1` measures between the shows.

Univariate and Multi-variate Plots

We can only assess the data so much without plots.

Generally, there are two things we want to understand quickly:

- distributions
- relationships
  - especially bi-variate or tri-variate relationships

Some background on `ggplot2`

Three major aspects:

- **layers**: The layers include all the points, bars, lines, etc. The `geom_` functions.
- **scales**: The scales the scales of the x and y, the colors, the fills, etc. The `scale_` functions.
- **facets**: The facets are the stratified plots. The `facet_` functions.

Additionally, the general look of the plot can be controlled by the `theme` functions.

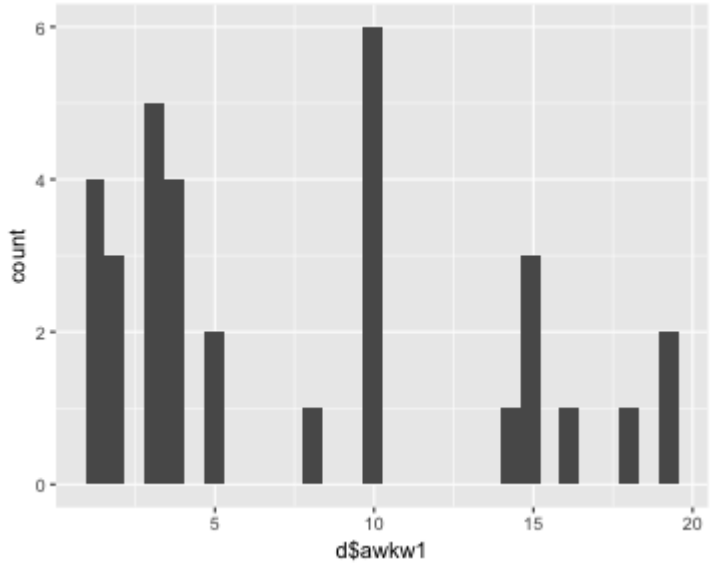
Univariate Plots

Simplest

```
qplot(d$awkw1)
```

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

```
## Warning: Removed 5 rows containing non-finite values (stat_bin).
```



Univariate Plots

Better

```
ggplot(d, aes(x=awkw1)) +
  geom_density(alpha = .5,
               fill = "dodgerblue4",
               color = "dodgerblue4")
```

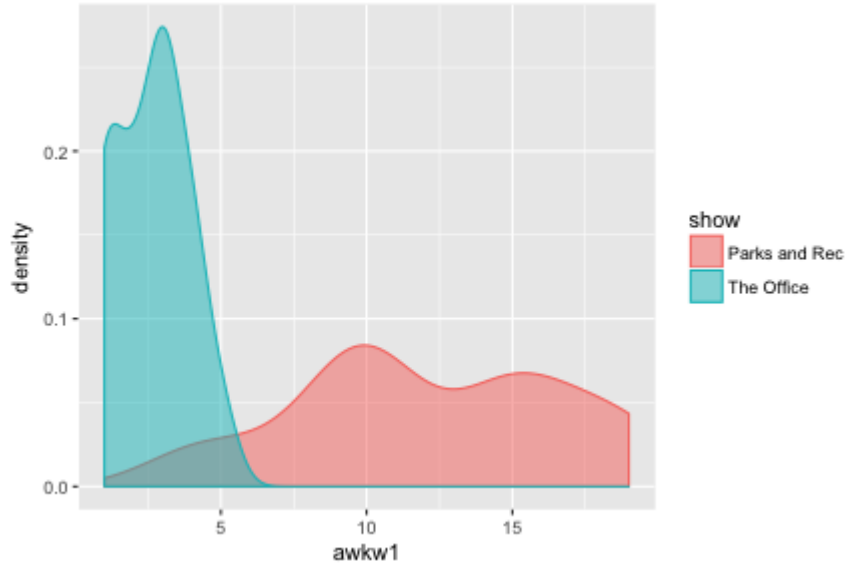
```
## Warning: Removed 5 rows containing non-finite values (stat_density).
```

Univariate-ish Plots

Even Better

```
ggplot(d, aes(x=awkw1, group = show, fill = show, color = show)) +
  geom_density(alpha = .5)
```

```
## Warning: Removed 5 rows containing non-finite values (stat_density).
```

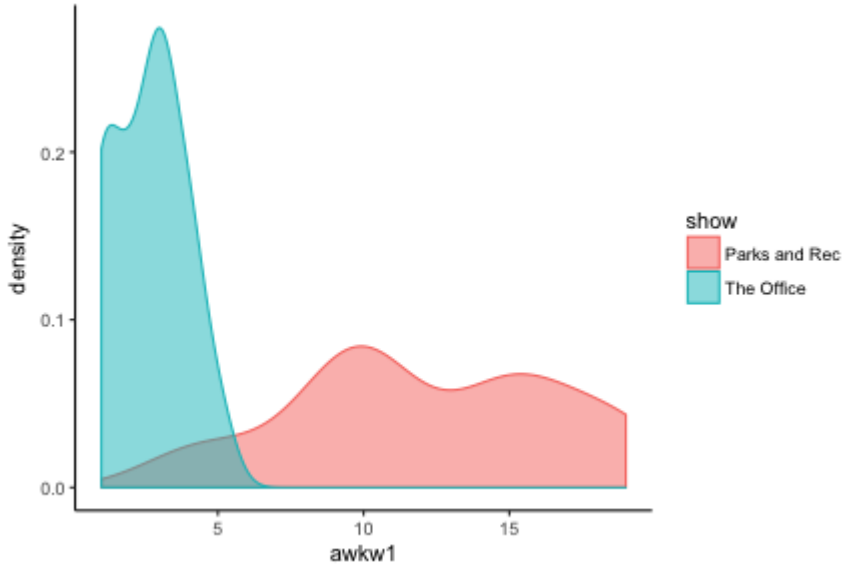


Univariate-ish Plots

Maybe Best?

```
ggplot(d, aes(x=awkw1, group = show, fill = show, color = show)) +
  geom_density(alpha = .5) +
  theme_classic()
```

```
## Warning: Removed 5 rows containing non-finite values (stat_density).
```



Some Notes, Comments, and Questions

- The warning: there are some missing values!
- Again, `awkw1` looks iffy.
- What else would you like to do to the plot?
- Would you want to try a different type of plot?

Multi-variate Plots

Let's check some relationships using

- scatterplots
- scatterplots with groups
- joy plots
- bar plots
- line plots

We'll use the `group`, `color` and `fill` options as well as the `facet_` functions.

Multi-variate Plots

Let's check some relationships using

- scatterplots
- scatterplots with groups
- joy plots
- bar plots
- line plots

We'll use the `group`, `color` and `fill` options as well as the `facet_` functions.

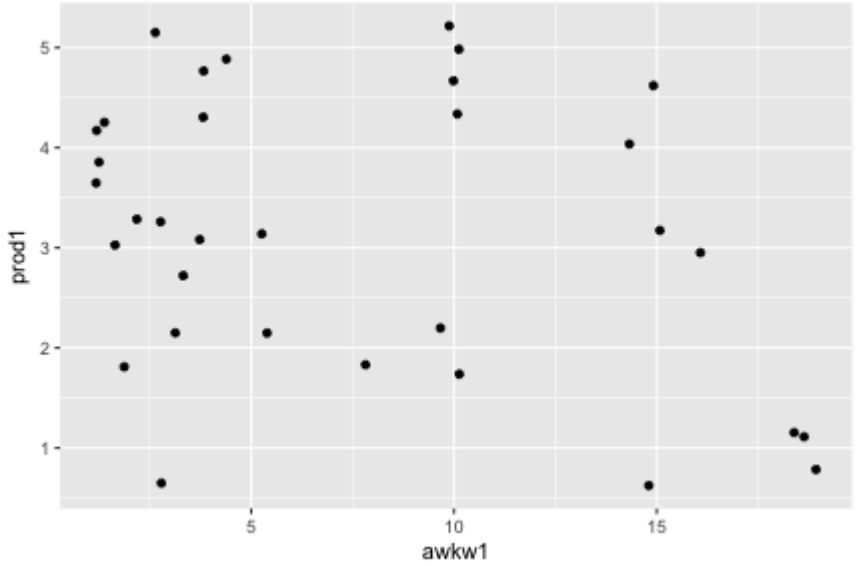
Let's get a blank plot started for each scatterplot and make `gend` a factor.

```
d$gend = factor(d$gend, labels = c("Male", "Female"))
scatter <- ggplot(d, aes(x = awkw1, y = prod1))
```

Scatterplot

```
scatter + geom_jitter()
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

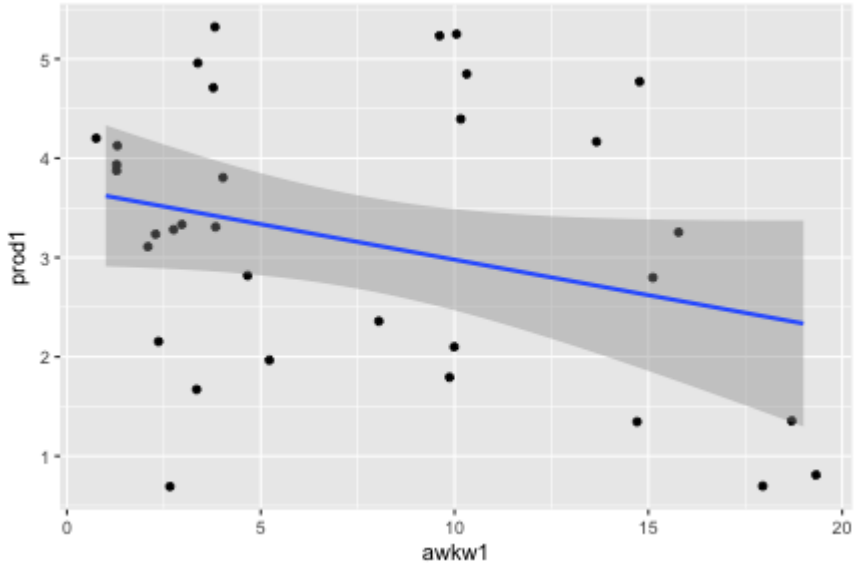


Scatterplot

```
scatter + geom_jitter() +
  geom_smooth(method = "lm")
```

```
## Warning: Removed 5 rows containing non-finite values (stat_smooth).
```

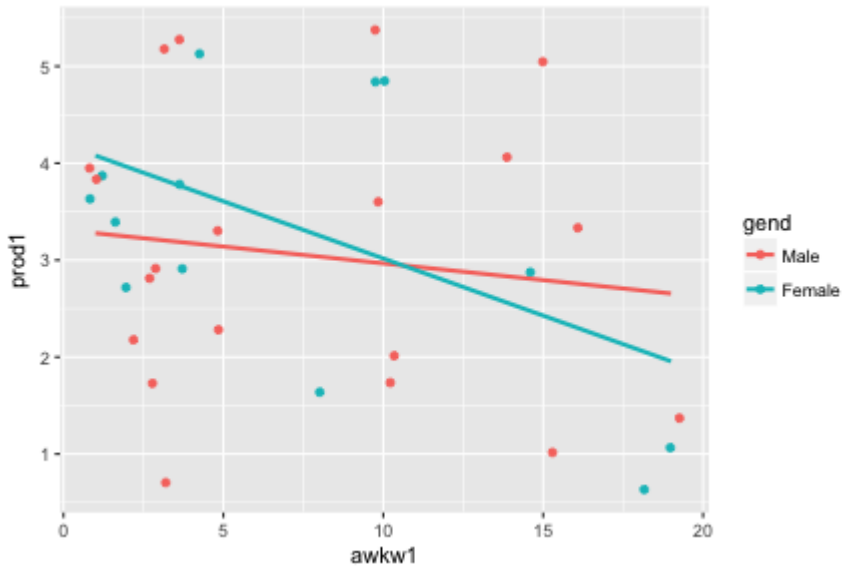
```
## Warning: Removed 5 rows containing missing values (geom_point).
```



Scatterplot with Interaction

Let's see if there appear to be differences across the genders.

```
scatter +
  geom_jitter(aes(group = gend, color = gend)) +
  geom_smooth(aes(group = gend, color = gend),
              method = "lm", se = FALSE)
```



Joy Plots

To create a joy plot, we need to reshape our data a bit. We are going to do two steps of reshaping:

1. Put our data in long form based on the time periods, and
2. Turn each variable into a single variable for the joy plot.

```
df_long = long(d,
               c("prod1", "prod2"),
               c("ment1", "ment2"),
               c("depr1", "depr2"),
               c("awkw1", "awkw2"),
               v.names = c("Prod", "Ment", "Depr", "Awkw"),
               timevar = "Time") %>%
  long(c("Prod", "Ment", "Depr", "Awkw"),
       v.names = "Value",
       timevar = "Variable",
       times = c("Prod", "Ment", "Depr", "Awkw"),
       id = c("id", "Time"))
```

df_long

What do you expect to see in `df_long`? Go step by step through the functions.

Copy CSV Excel PDF Print Search:

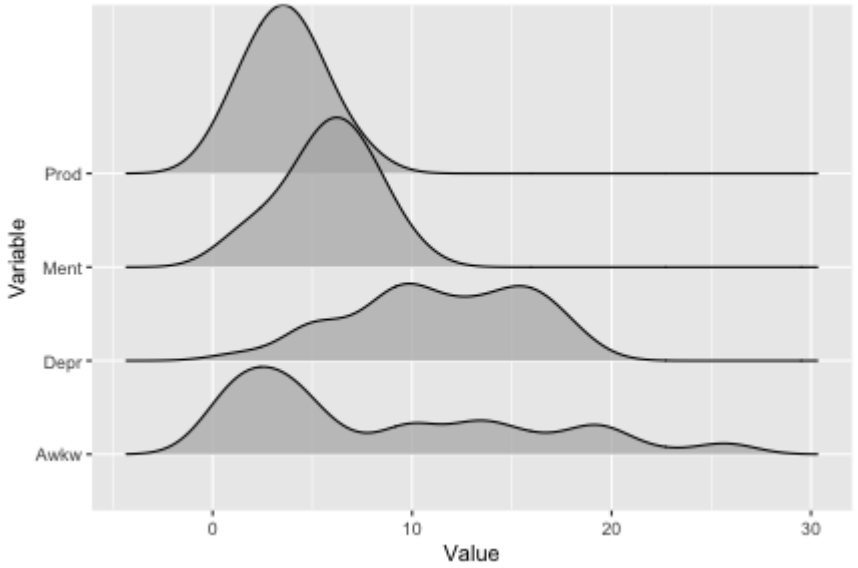
	nam	phys	marr	gend	race	inco	chil
1.1.Prod	Michael	8	0	Male	White	55	0
2.1.Prod	Pam	7	1	Female	White	35	2
3.1.Prod	Jim	8	1	Male	White	70	2
4.1.Prod	Dwight	8	0	Male	White	70	0
5.1.Prod	Stanley	4	1	Male	Black	70	1
6.1.Prod	Phyllis	4	1	Female	White	70	0

Showing 1 to 6 of 304 entries

Previous 1 2 3 4 5 … 51 Next

Joy Plots

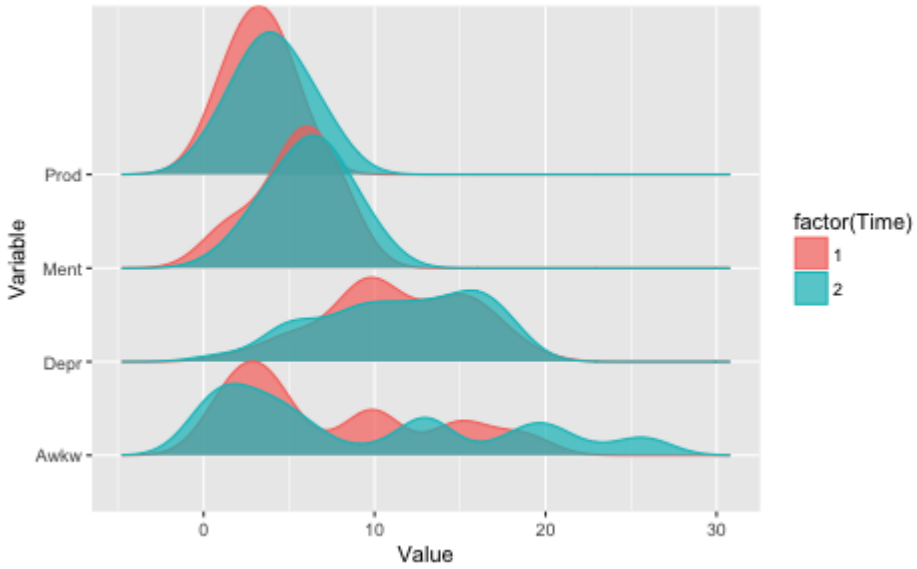
```
library(ggjoy)
ggplot(df_long, aes(x = Value, y = Variable)) +
  geom_joy(alpha = .7)
```



This highlights the overall distributions but ignores time points.

Joy Plots

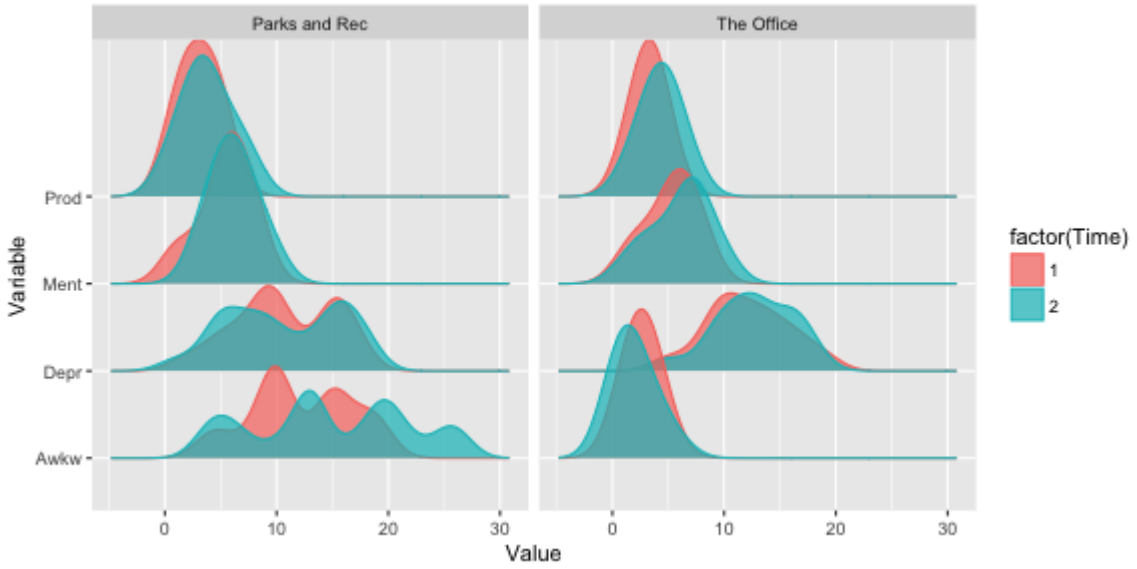
```
ggplot(df_long, aes(x = Value, y = Variable)) +
  geom_joy(aes(fill = factor(Time),
               color = factor(Time)),
           alpha = .7)
```



This, however, ignores the differences by show.

Joy Plots

```
ggplot(df_long, aes(x = Value, y = Variable)) +
  geom_joy(aes(fill = factor(Time),
               color = factor(Time)),
           alpha = .7) +
  facet_grid(~show, scales = "free")
```



What patterns do you see?

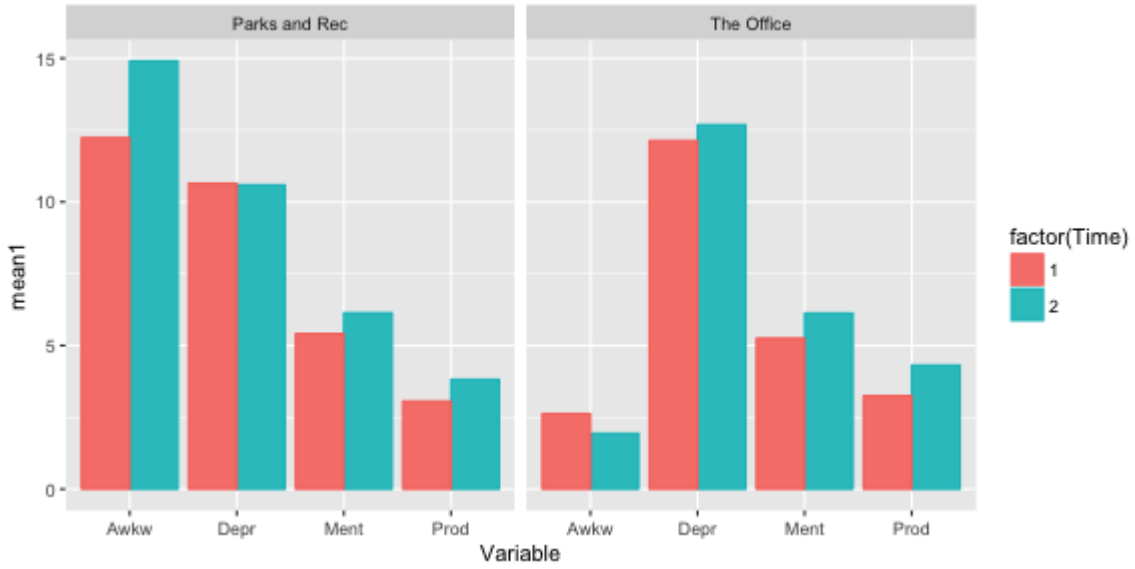
Bar Plots

In general, to do bar plots, we want some summary statistics (e.g., means and standard errors).

```
summed = df_long %>%
  group_by(Time, Variable, show) %>%
  summarize(mean1 = mean(Value, na.rm=TRUE),
            se1   = sd(Value, na.rm=TRUE)/sqrt(n()))
```

Bar Plots

```
ggplot(summed, aes(x = Variable, y = mean1)) +
  geom_bar(stat = "identity", position = "dodge",
           aes(fill = factor(Time),
               color = factor(Time)),
           alpha = .9) +
  facet_grid(~show, scales = "free")
```



This is probably a bit much... Let's try a line plot instead.

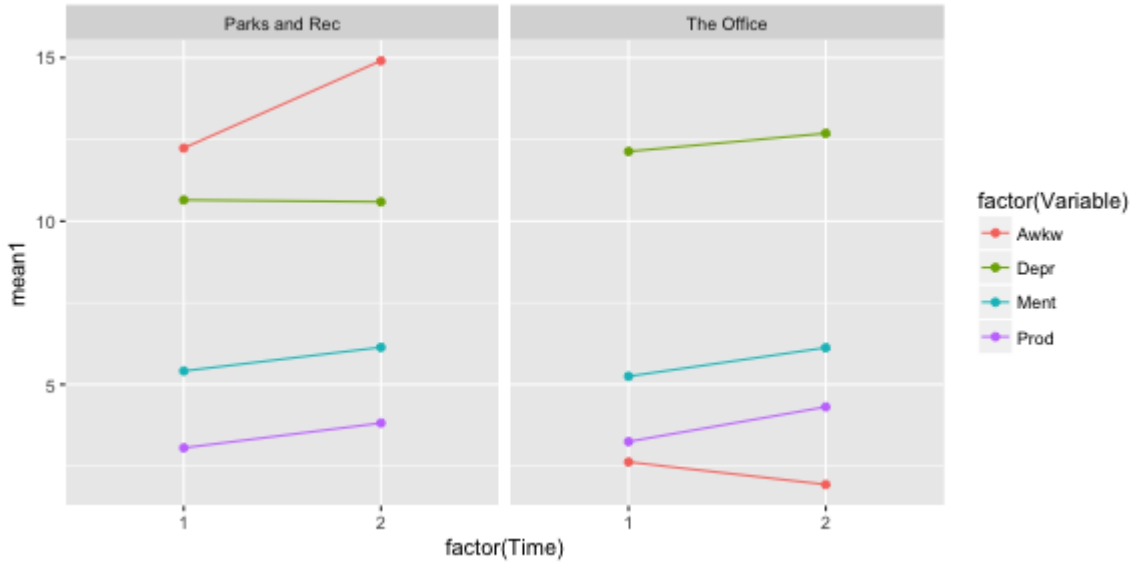
Line Plots

We are going to look at two types

1. Pre-Post Plots
2. Spaghetti plots

Pre-Post Plots

```
ggplot(summed, aes(x = factor(Time), y = mean1,
                   group = factor(Variable), color = factor(Variable)
  geom_line(alpha = .9) +
  geom_point() +
  facet_grid(~show, scales = "free")
```

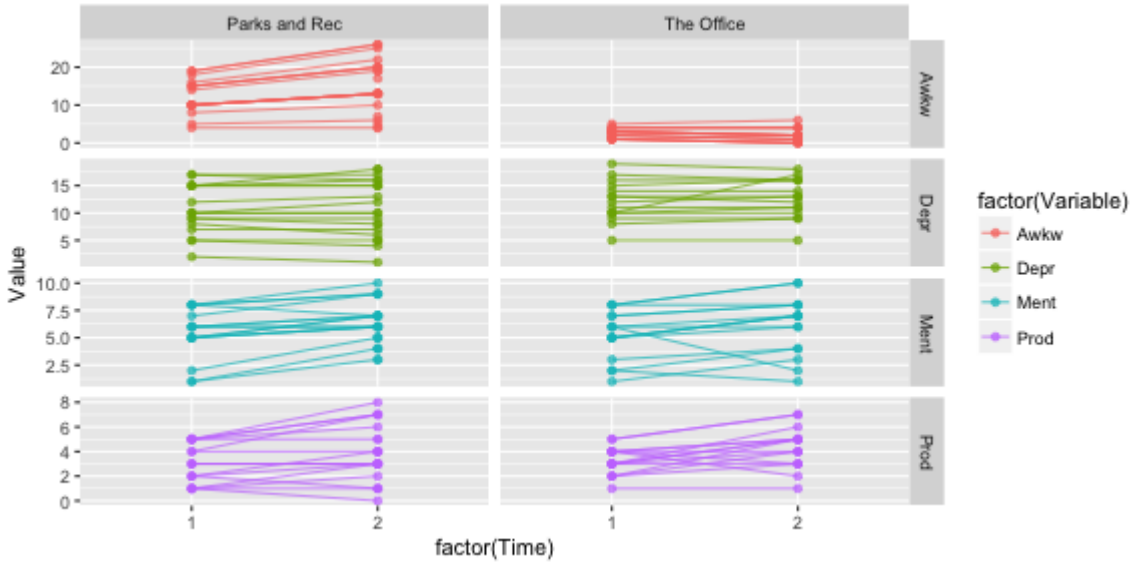


Much better! We can see trends for each variable across both time points really easily.

Spaghetti Plot

We may want to see individual trajectories. Spaghetti plots are made for this.

```
ggplot(df_long, aes(x = factor(Time), y = Value,
                    group = interaction(Variable, nam), color = factor
  geom_line(alpha = .7) +
  geom_point(alpha = .7) +
  facet_grid(Variable~show, scales = "free")
```



Spaghetti Plot

I want to highlight a piece of the code before.

```
ggplot(df_long, aes(x = factor(Time), y = Value,
                    group = interaction(Variable, nam) , color = fact
  geom_line(alpha = .7) +
  geom_point(alpha = .7) +
  facet_grid(Variable~show, scales = "free")
```

Note the:

- `interaction(Variable, nam)`
- `Variable~show` in the `facet_grid()` function

`interaction()` let's us group by more than one variable. Why would we want to do that here?

`Variable~show` facets the plots by both variables where one is the rows (`Variable`) and one is the columns (`show`). We could have them both be the rows (`Variable+show~.`) or both be the columns (`~Variable+show`).

Use it 💪

Using the "OfficeParks" data set:

- **Understand time trends**
- **Find out if `awkw2` has the same problem as `awkw1`**
- **Find a way to fix `awkw1` (and `awkw2` if necessary)**
- **Demonstrate where there is a strong bivariate relationship (using a plot)**

What did you find??