A decorative network graph pattern in the top-left corner, featuring a complex web of nodes and edges. Some nodes are highlighted with blue circles or dots, while others are grey. The edges are thin grey lines.

ggplot

Making data visualizations beautiful

A decorative network graph pattern in the bottom-right corner, similar to the one in the top-left, with nodes and edges, some highlighted in blue.

ggplot::common visualizations

- ◎ Boxplots (`geom_boxplot`)
- ◎ Histograms (`geom_histogram`)
- ◎ Barplots (`geom_bar`)
- ◎ Line charts (`geom_line`)
- ◎ Scatterplots (`geom_point`)
- ◎ And more!

All ggplot visualizations work similarly to one another. The concepts in this session focus on bar plots but can be applied to other ggplot visualizations. You can find helpful documentation on how to build other ggplot visualizations here:

<http://rstudio-connect.chop.edu/connect/#/apps/116/access/44>

ggplot::general format

```
data %>%  
  ggplot(aes(x=variable_on_xaxis,  
             y = variable_on_yaxis,  
             fill = group_to_fill_in)) +  
  geom_bar(stat = 'identity') +  
  labs(x = 'x axis label',  
       y = 'y axis label',  
       title = 'informative title of this graph')
```

ggplot::general format

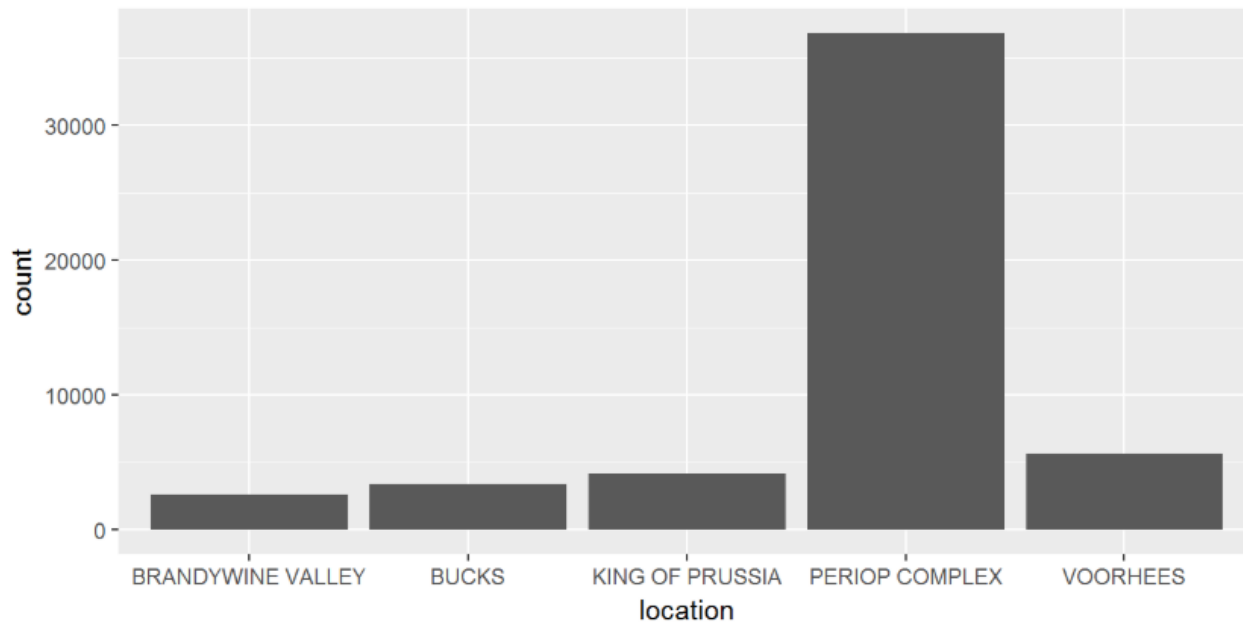
```
data %>%  
  ggplot(aes(x=variable_on_xaxis,  
             y = variable_on_yaxis,  
             fill = group_to_fill_in)) +  
  geom_bar(stat = 'identity') +  
  labs(x = 'x axis label',  
       y = 'y axis label',  
       title = 'informative title of this graph')
```

Is this confusing? No worries, we have examples!

ggplot::bar chart example

```
# ready data by aggregating cases by CHOP | location
loc <- cases %>%
  dplyr::group_by(location) %>%
  dplyr::summarise(count = n())

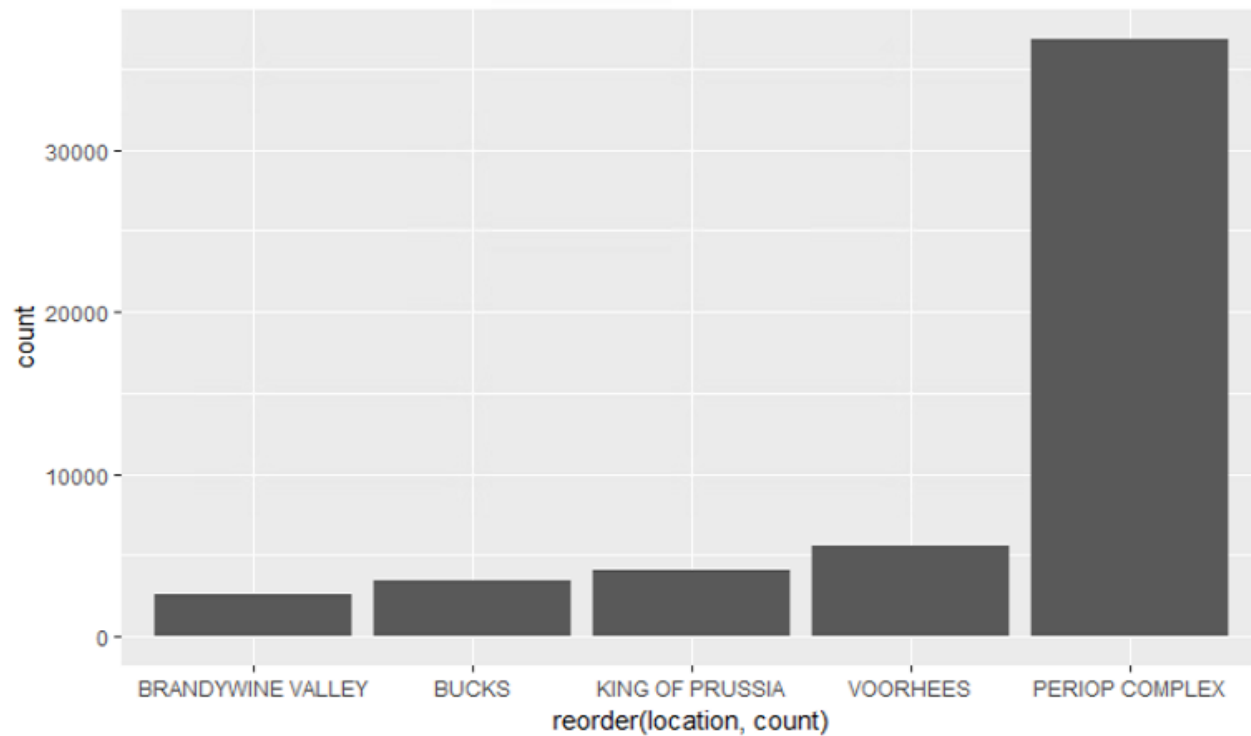
ggplot(loc, aes(x=location, y=count)) +
  geom_bar(stat="identity") #use stat = 'identity' when data is pre-aggregated
```



ggplot::bar chart example

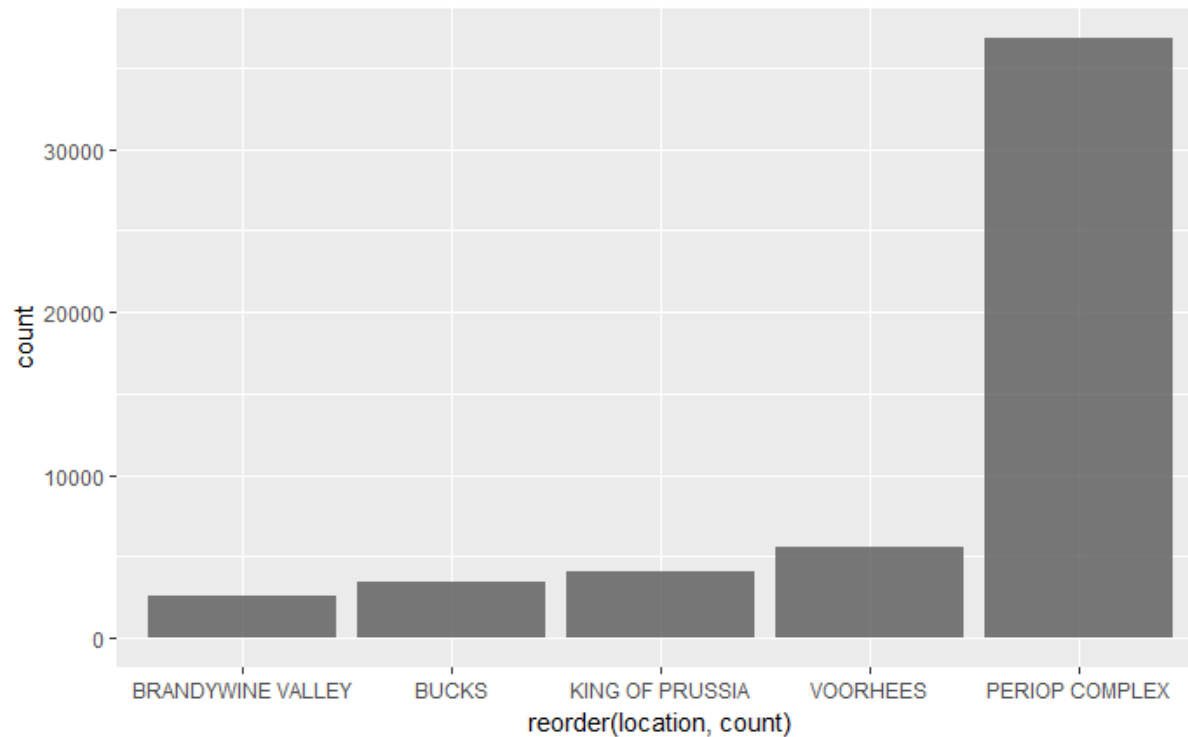


```
ggplot(loc, aes(x=reorder(location, count), y=count)) + #reorder categories in order of count  
  geom_bar(stat="identity")
```



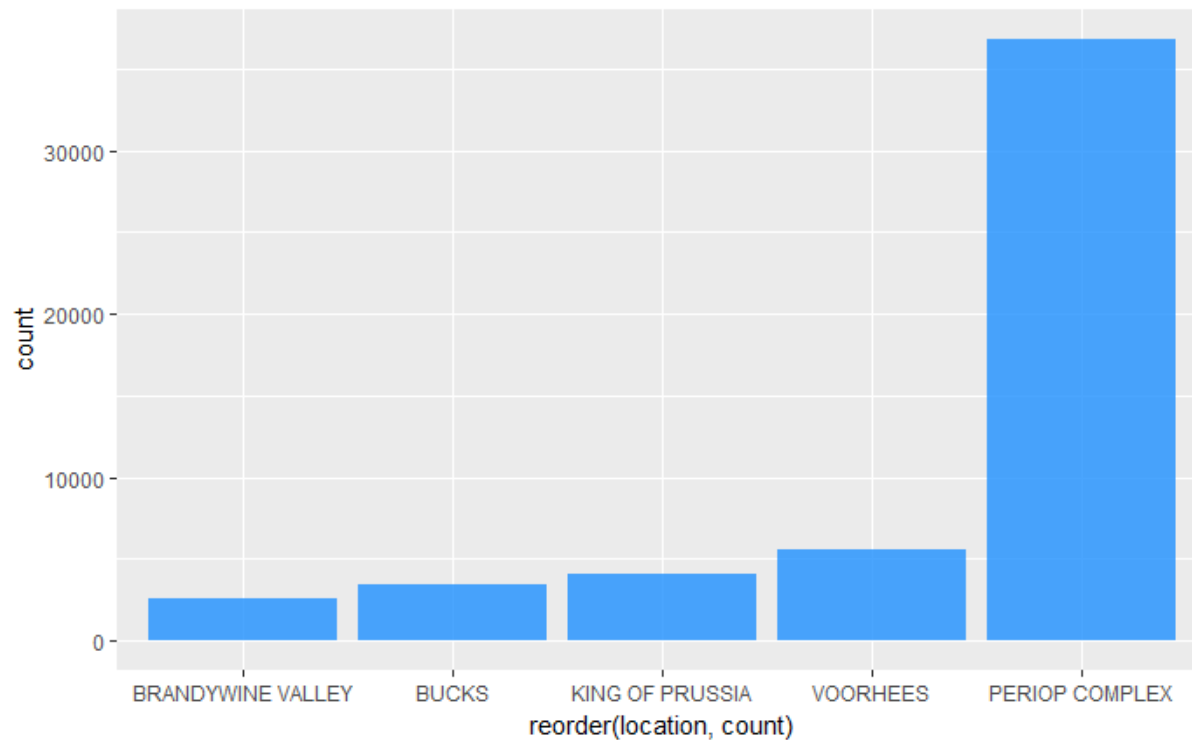
ggplot::bar chart example

```
ggplot(loc, aes(x=reorder(location, count), y=count)) +  
  geom_bar(stat="identity", alpha = .8) #alpha makes bars somewhat transparent
```



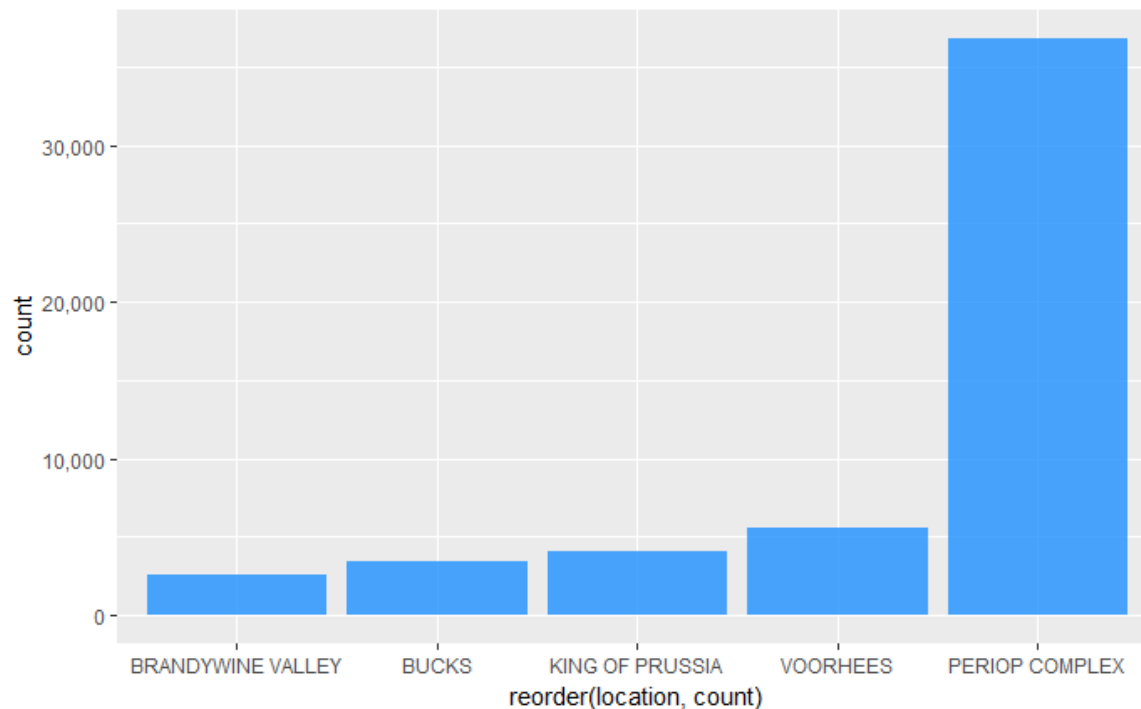
ggplot::bar chart example

```
ggplot(loc, aes(x=reorder(location, count), y=count)) +  
  geom_bar(stat="identity", alpha = .8, fill = "dodgerblue") #fills your bars with beautiful colors
```



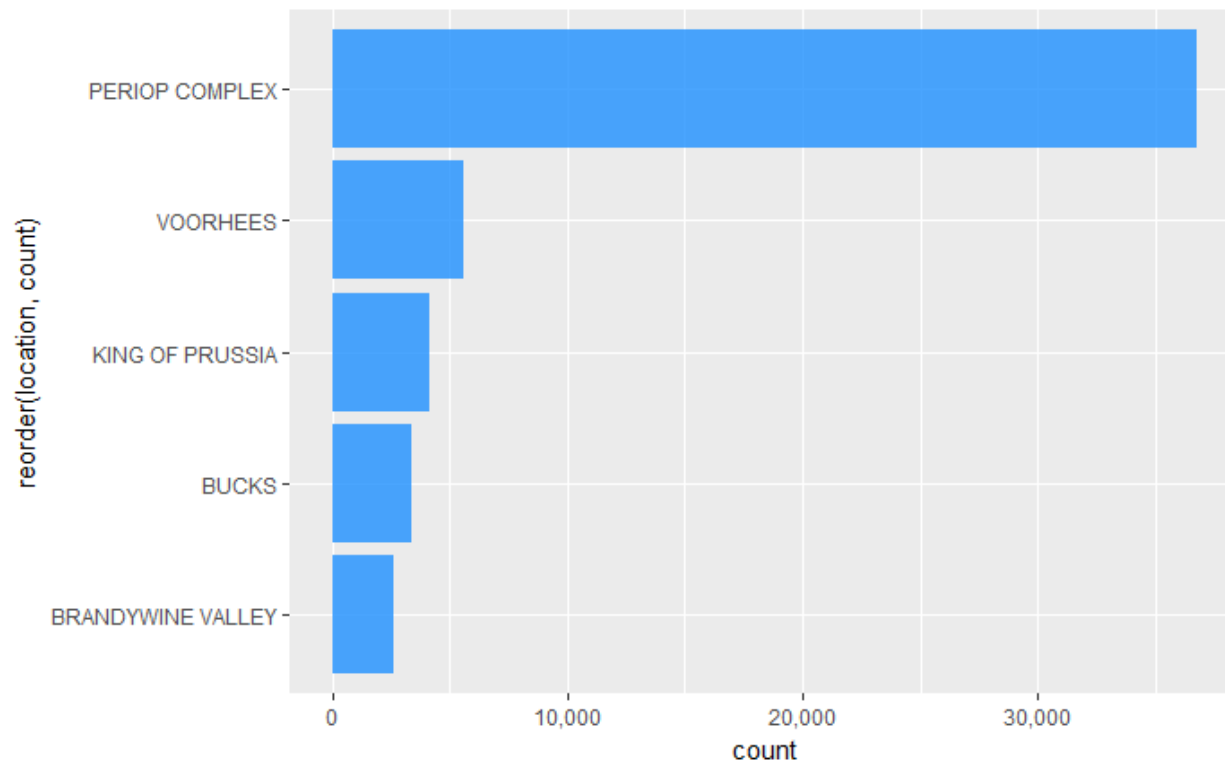
ggplot::bar chart example

```
library(scales)
ggplot(loc, aes(x=reorder(location, count), y=count)) +
  geom_bar(stat="identity", alpha = .8, fill = "dodgerblue") +
  scale_y_continuous(labels=comma) # from the scale package - will put commas in your numbers
                                     # can also yuse this package to specify percents or currency!
```



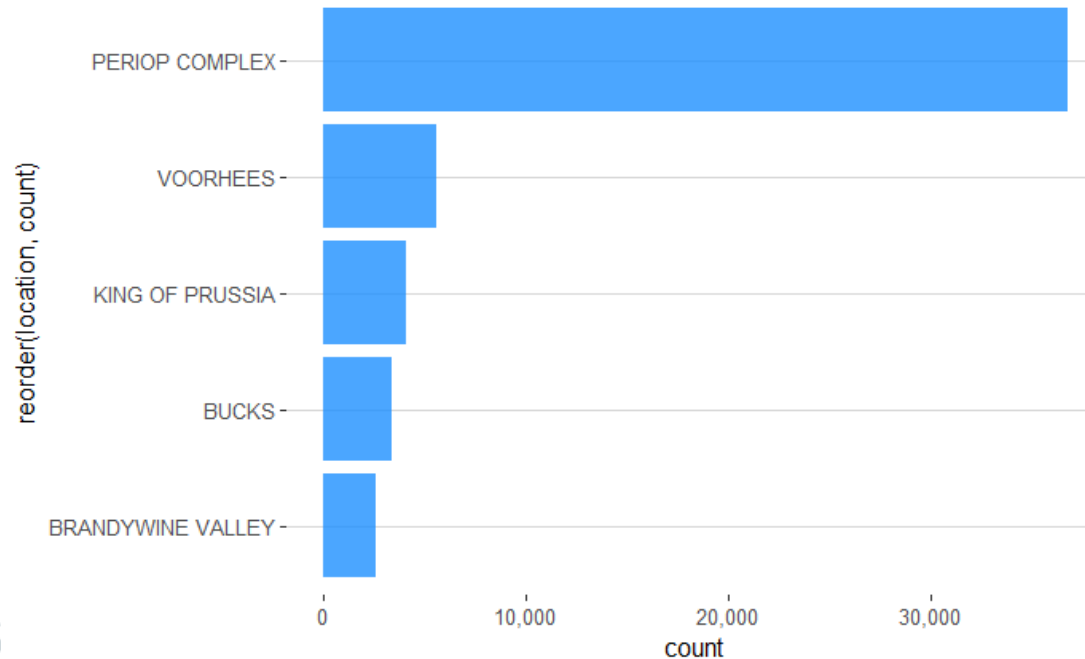
ggplot::bar chart example

```
ggplot(loc, aes(x=reorder(location, count), y=count)) +  
  geom_bar(stat="identity", alpha = .8, fill = "dodgerblue") +  
  scale_y_continuous(labels=comma) +  
  coord_flip() # flip your axes!
```



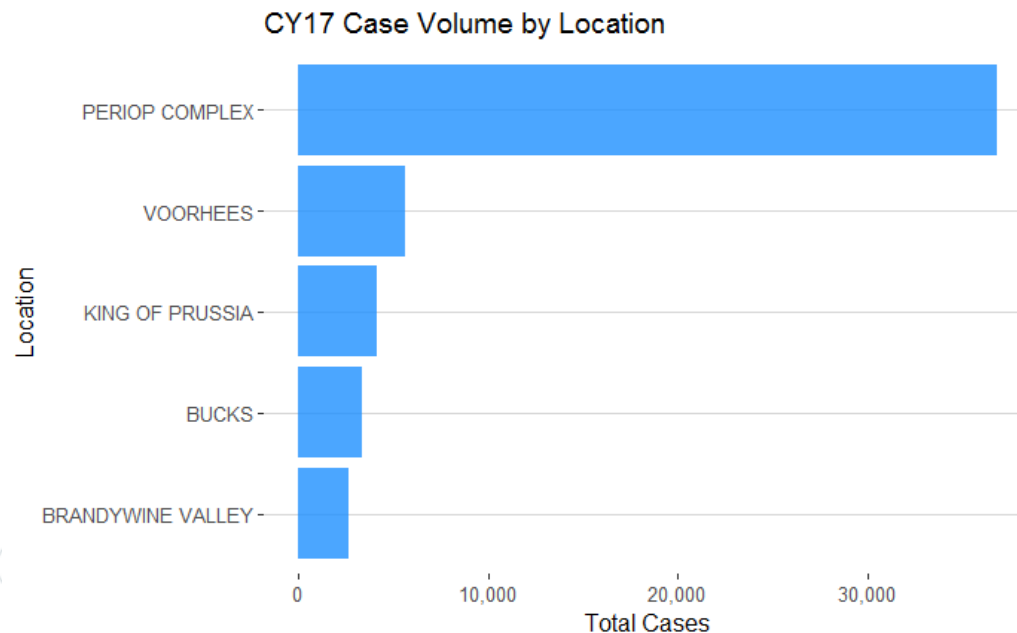
ggplot::bar chart example

```
ggplot(loc, aes(x=reorder(location, count), y=count)) +  
  geom_bar(stat="identity", alpha = .8, fill = "dodgerblue") +  
  scale_y_continuous(labels=comma) +  
  coord_flip() +  
  ggthemes::theme_hc() #apply a pre-loaded style theme!
```



ggplot::bar chart example

```
ggplot(loc, aes(x=reorder(location, count), y=count)) +  
  geom_bar(stat="identity", alpha = .8, fill = "dodgerblue") +  
  scale_y_continuous(labels=comma) +  
  coord_flip() +  
  ggthemes::theme_hc() +  
  #add informative labels!  
  labs(x= "Location",  
        y = "Total Cases",  
        title = "CY17 Case Volume by Location")
```

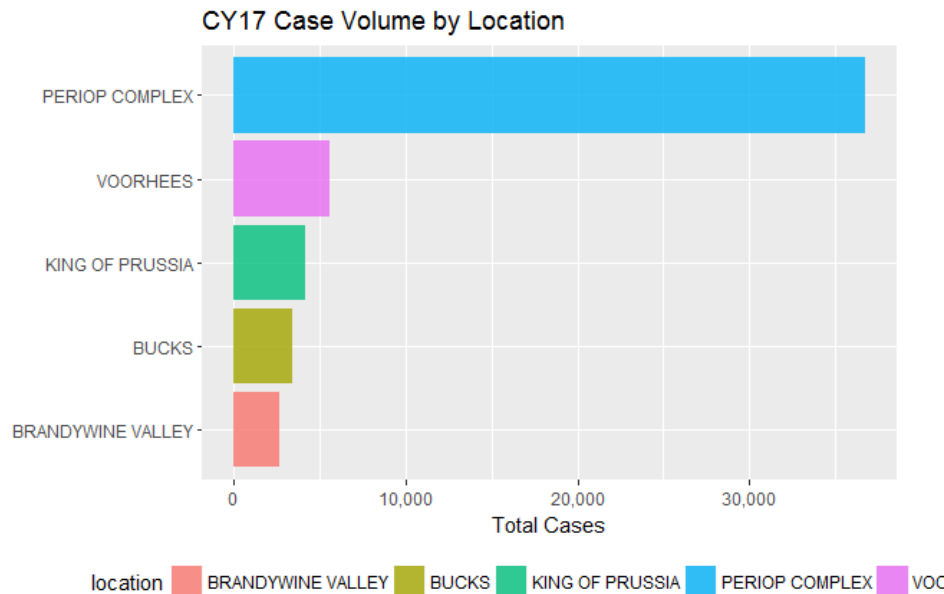


A decorative background featuring a network diagram. It consists of numerous nodes, represented by circles of varying sizes and shades of gray, some of which are highlighted with blue outlines. These nodes are interconnected by a web of thin, light gray lines, creating a complex, organic structure that resembles a molecular or social network. The diagram is positioned in the corners of the slide, framing the central text.

**Whoa, that was fun! Let's get
more complicated!**

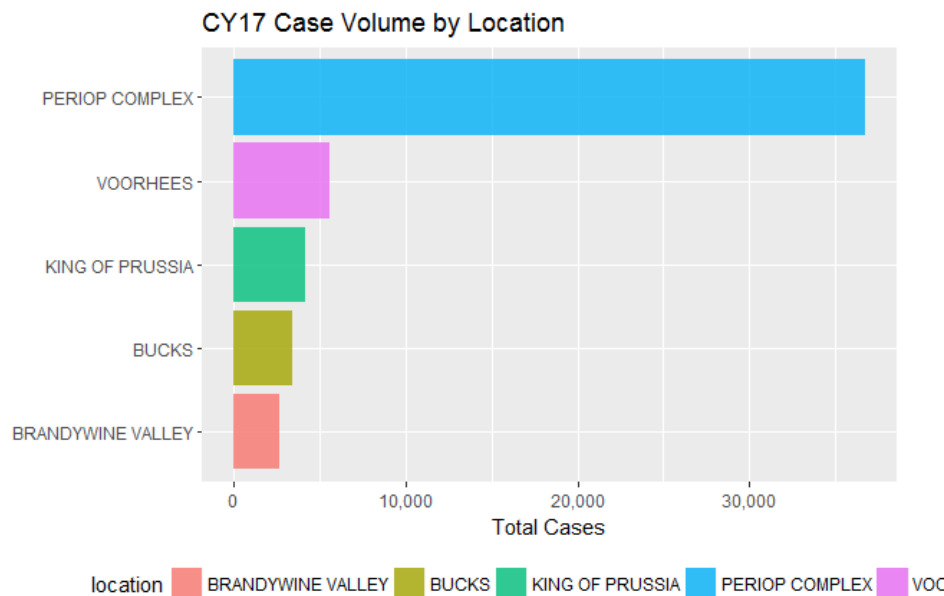
Bar charts: colors and legends

```
ggplot(loc, aes(x=reorder(location, count), y=count, fill = location)) + # fill based on group
  geom_bar(stat="identity", alpha = .8) +
  scale_y_continuous(labels=scales::comma) +
  theme(aspect.ratio = 1/1.6, # maintains x-axis:y-axis ratio
        legend.position = "bottom") + # puts a legend on the bottom of your graph
  coord_flip() +
  labs(x= "",
       y = "Total Cases",
       title = "CY17 Case Volume by Location")
```

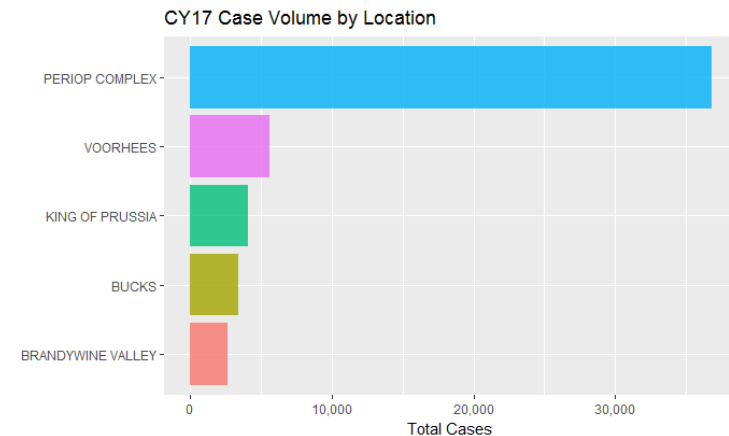


Bar charts: colors and legends

```
ggplot(loc, aes(x=reorder(location, count), y=count, fill = location)) + # fill based on group
  geom_bar(stat="identity", alpha = .8) +
  scale_y_continuous(labels=scales::comma) +
  theme(aspect.ratio = 1/1.6, # maintains x-axis:y-axis ratio
        legend.position = "bottom") + # puts a legend on the bottom of your graph
  coord_flip() +
  labs(x= "",
       y = "Total Cases",
       title = "CY17 Case Volume by Location")
```



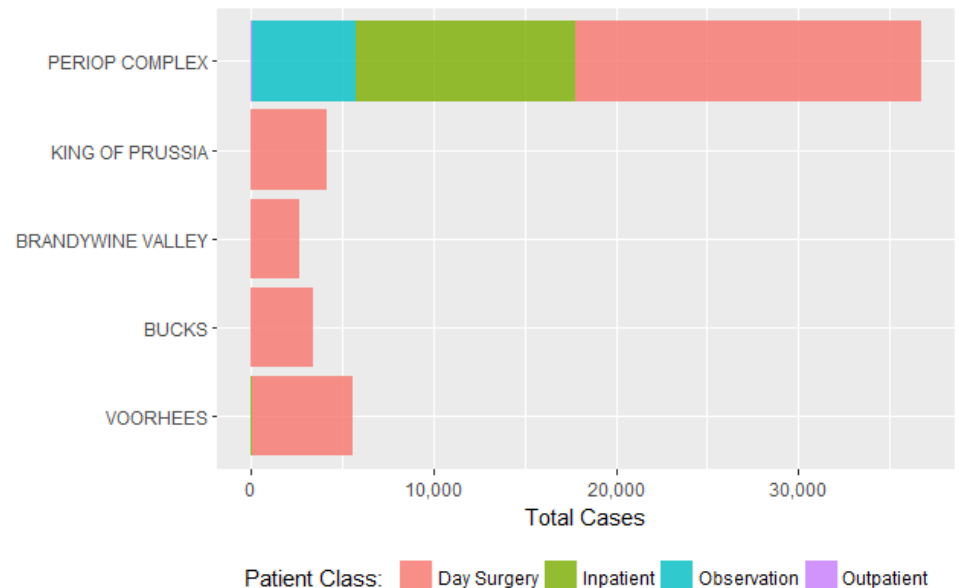
Don't not want a legend?
Use: `legend.position = "none"`



Bar chart groups: stacked vs dodged

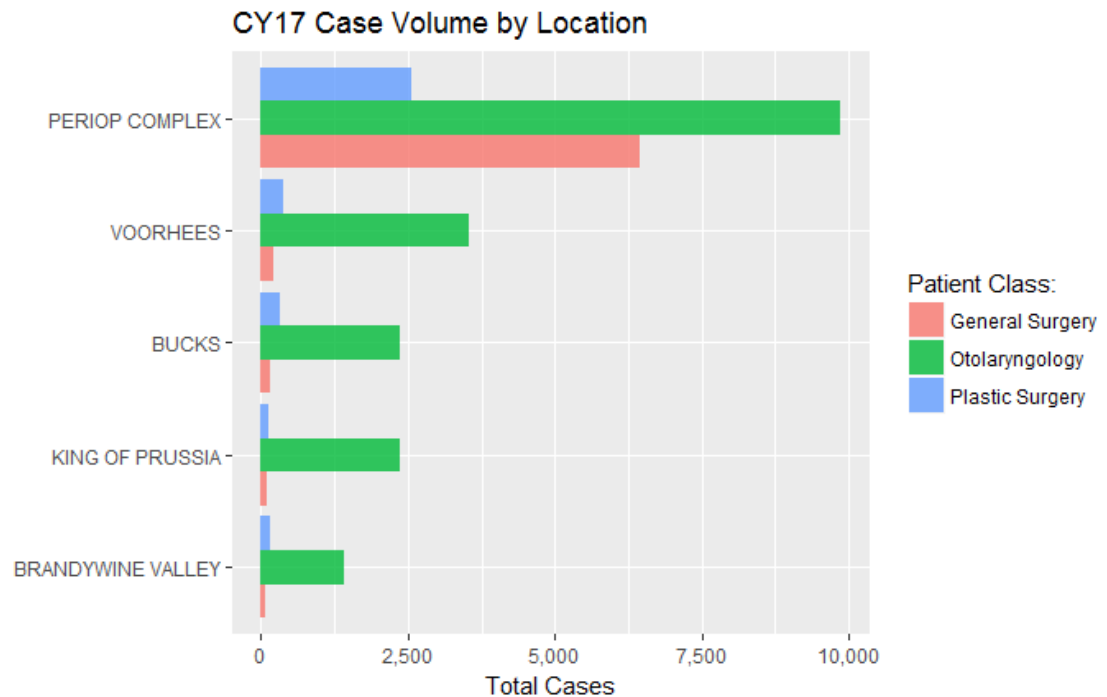
```
pat_class <- cases %>%  
  group_by(location, patient_class) %>%  
  summarise(count = n())  
  
ggplot(pat_class, aes(x=reorder(location, count), y=count, fill = patient_class)) +  
  geom_bar(stat="identity", alpha = .8, position = "stack") + #stack your barchart  
  scale_y_continuous(labels=scales::comma) +  
  coord_flip() +  
  theme(aspect.ratio = 1/1.6, legend.position = 'bottom') +  
  labs(x= " ",  
       y = "Total Cases",  
       title = "CY17 Case Volume by Location") +  
  guides(fill=guide_legend(title="Patient Class: "))
```

CY17 Case Volume by Location



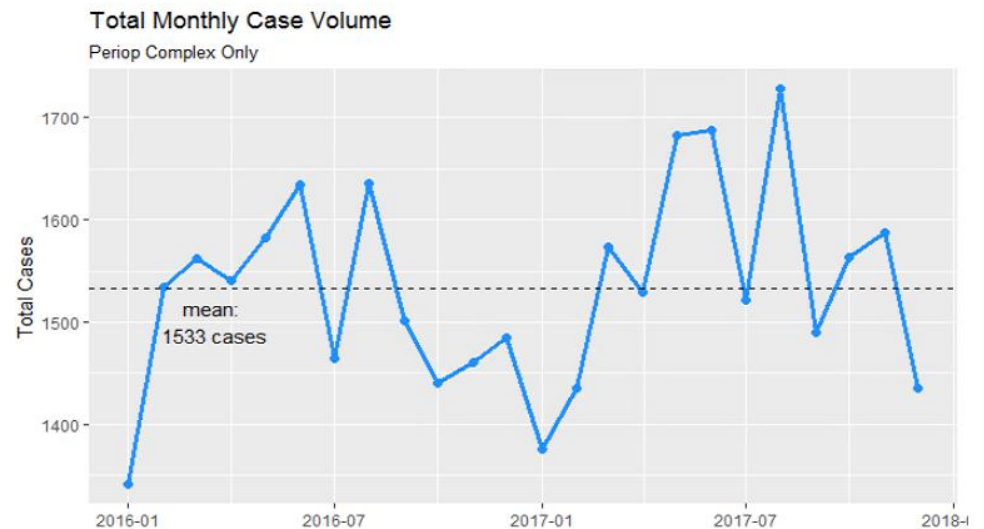
Bar chart groups: stacked vs dodged

```
svc <- cases %>%  
  filter(service %in% c("General Surgery", "Otolaryngology", "Plastic Surgery")) %>%  
  group_by(location, service) %>%  
  summarise(count = n())  
  
ggplot(svc, aes(x=reorder(location, count), y=count, group = service, fill = service)) +  
  geom_bar(stat='identity', alpha = .8, position = "dodge") + # make bars side-by-side  
  scale_y_continuous(labels=comma) +  
  coord_flip() +  
  labs(x= " ",  
       y = "Total Cases",  
       title = "CY17 Case Volume by Location") +  
  guides(fill=guide_legend(title="Patient Class: "))
```



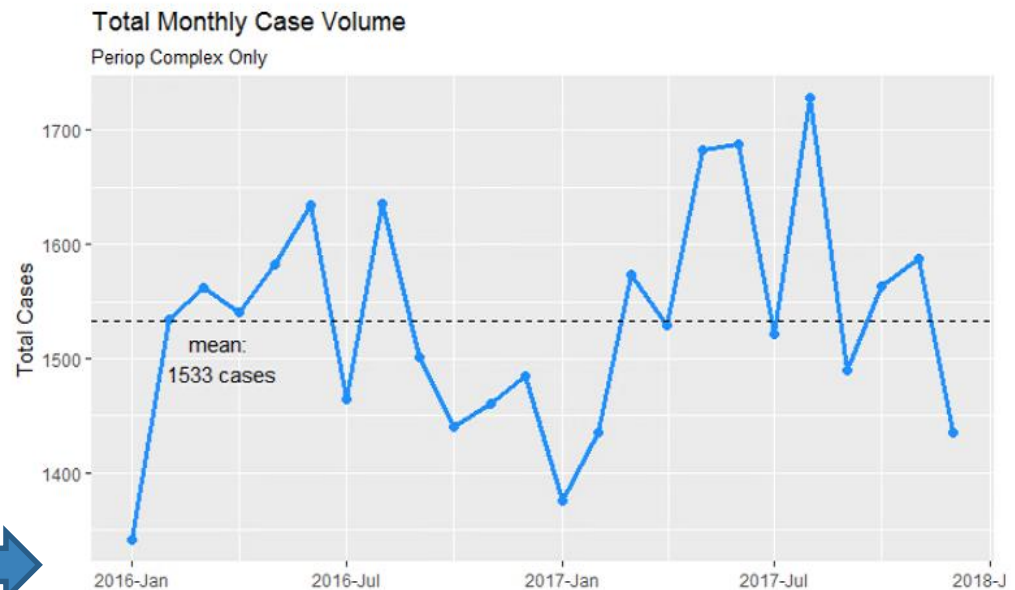
Line charts, oh my!

```
main_monthly <- cases %>%  
  filter(location == 'PERIOP COMPLEX') %>%  
  mutate(month_year = floor_date(surgery_date, "month")) %>%  
  group_by(month_year) %>%  
  summarise(count = n()) %>%  
  mutate(avg_cases = round(mean(count), 0))  
  
ggplot(main_monthly, aes(x=month_year, y=count)) +  
  geom_line(size = 1.2, color = "dodgerblue") + # makes line graph over time  
  geom_point(size=2, color = "dodgerblue") + # adds point to your line  
  geom_hline(aes(yintercept = avg_cases), color = "black", linetype = 'dashed') +  
  annotate("text", x = as.Date("2016-03-15"), y = 1500,  
    label = paste("mean:\n", unique(main_monthly$avg_cases), "cases", sep=" ")) +  
  labs(x=" ",  
    y = "Total Cases",  
    title = "Total Monthly Case Volume",  
    subtitle = "Periop Complex Only")
```



Line charts, oh my!

```
main_monthly <- cases %>%  
  filter(location == 'PERIOP COMPLEX') %>%  
  mutate(month_year = floor_date(surgery_date, "month")) %>%  
  group_by(month_year) %>%  
  summarise(count = n()) %>%  
  mutate(avg_cases = round(mean(count), 0))  
  
ggplot(main_monthly, aes(x=month_year, y=count)) +  
  geom_line(size = 1.2, color = "dodgerblue") + # makes line graph over time  
  geom_point(size=2, color = "dodgerblue") + #adds point to your line  
  geom_hline(aes(yintercept = avg_cases), color = "black", linetype = 'dashed') +  
  annotate("text", x = as.Date("2016-03-15"), y = 1500,  
    label = paste("mean:\n", unique(main_monthly$avg_cases), "cases", sep=" ")) +  
  scale_x_date(labels = date_format('%Y-%b')) +  
  labs(x=" ",  
    y = "Total Cases",  
    title = "Total Monthly Case Volume",  
    subtitle = "Periop Complex Only")
```



ggplot::exercises

- 1) Create a barplot of number of central vs. non central line blood cultures in FY17
- 2) Create a stacked barplot by patients under 90 days and patients over 90 days for the graph you just created (central vs non-central lines)
- 3) Create a side-by-side barplot of graph #2
- 4) Create a line graph of counts of blood cultures over time (months)

