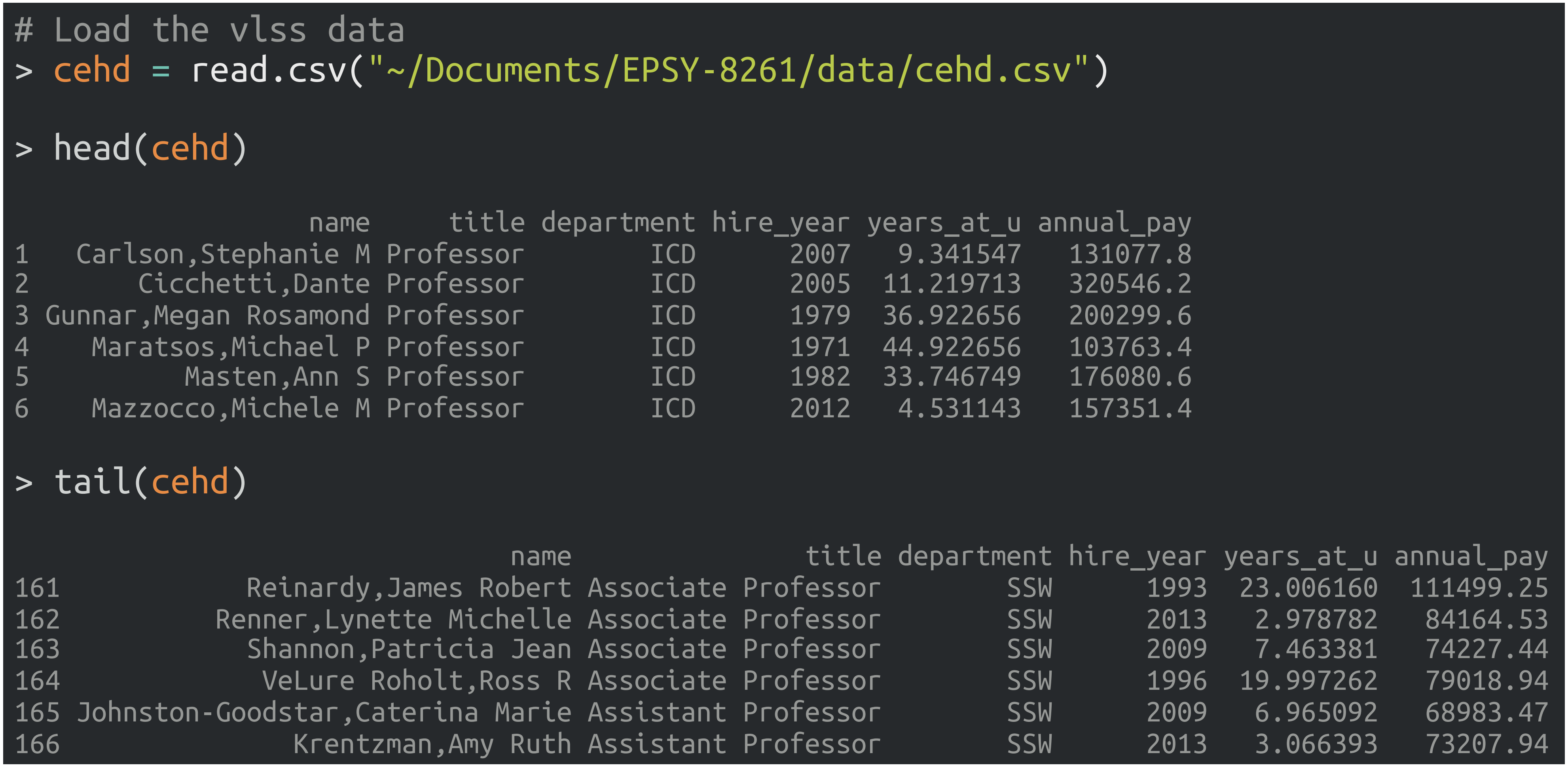
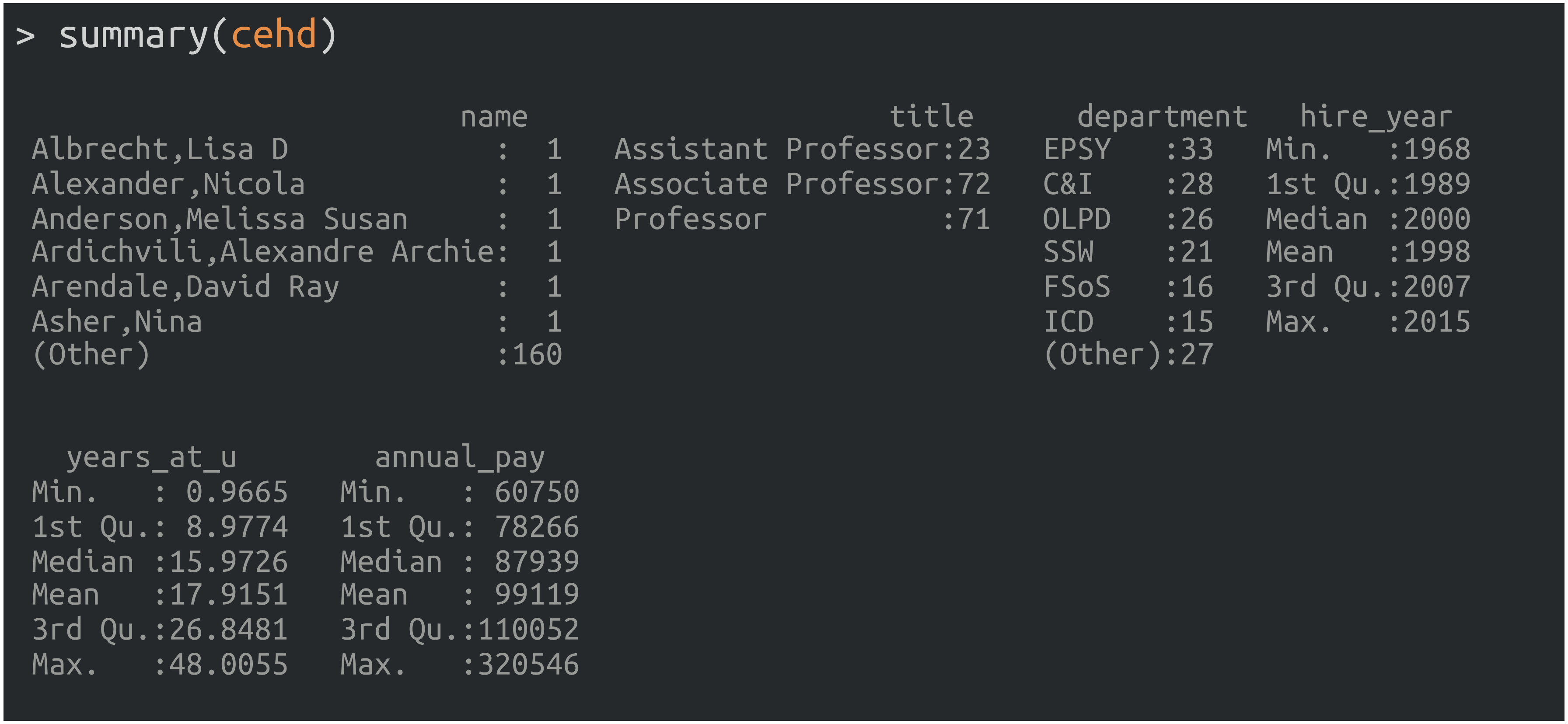
Plotting with ggplot2

Andrew Zieffler



```
# Load the vlss data
> cehd = read.csv("~/Documents/EPSY-8261/data/cehd.csv")

> head(cehd)

                   name     title department hire_year years_at_u annual_pay
1   Carlson,Stephanie M Professor        ICD      2007   9.341547   131077.8
2       Cicchetti,Dante Professor        ICD      2005  11.219713   320546.2
3 Gunnar,Megan Rosamond Professor        ICD      1979  36.922656   200299.6
4    Maratsos,Michael P Professor        ICD      1971  44.922656   103763.4
5          Masten,Ann S Professor        ICD      1982  33.746749   176080.6
6    Mazzocco,Michele M Professor        ICD      2012   4.531143   157351.4

> tail(cehd)

                                name               title department hire_year years_at_u annual_pay
161            Reinardy,James Robert Associate Professor        SSW      1993  23.006160  111499.25
162          Renner,Lynette Michelle Associate Professor        SSW      2013   2.978782   84164.53
163            Shannon,Patricia Jean Associate Professor        SSW      2009   7.463381   74227.44
164             VeLure Roholt,Ross R Associate Professor        SSW      1996  19.997262   79018.94
165 Johnston-Goodstar,Caterina Marie Assistant Professor        SSW      2009   6.965092   68983.47
166               Krentzman,Amy Ruth Assistant Professor        SSW      2013   3.066393   73207.94
```

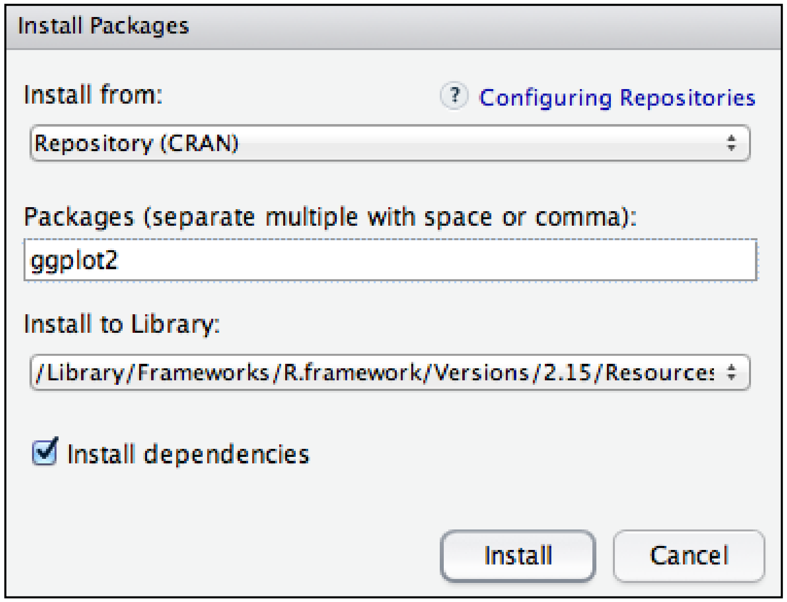


```
> summary(cehd)

                          name                         title      department   hire_year   
 Albrecht,Lisa D            :  1   Assistant Professor:23   EPSY   :33   Min.   :1968  
 Alexander,Nicola           :  1   Associate Professor:72   C&I    :28   1st Qu.:1989  
 Anderson,Melissa Susan     :  1   Professor          :71   OLPD   :26   Median :2000  
 Ardichvili,Alexandre Archie:  1                            SSW    :21   Mean   :1998  
 Arendale,David Ray         :  1                            FSoS   :16   3rd Qu.:2007  
 Asher,Nina                 :  1                            ICD    :15   Max.   :2015  
 (Other)                    :160                            (Other):27                 


   years_at_u        annual_pay    
 Min.   : 0.9665   Min.   : 60750  
 1st Qu.: 8.9774   1st Qu.: 78266  
 Median :15.9726   Median : 87939  
 Mean   :17.9151   Mean   : 99119  
 3rd Qu.:26.8481   3rd Qu.:110052  
 Max.   :48.0055   Max.   :320546  
```

Install the **ggplot2** Package

Using the RStudio GUI...

- Click the **Packages** tab.
- Click **Install Packages**.
- Enter *ggplot2* in the text box.
- Click **Install**.



Install Packages

Install from: Repository (CRAN)

Packages (separate multiple with space or comma): ggplot2

Install to Library: /Library/Frameworks/R.framework/Versions/2.15/Resource

☑ Install dependencies

...or directly from the R command line...

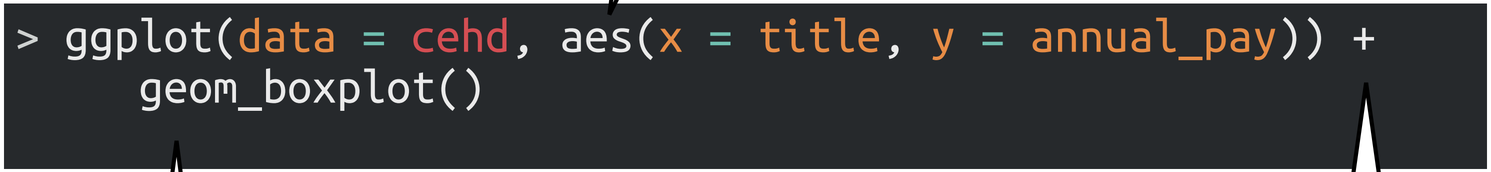
```
> install.packages(“ggplot2”, dependencies = TRUE)
```

The `library()` function loads the package so that the functions in the package are accessible. Libraries need to be loaded *every* R session.

```
# Load the ggplot2 library
> library(ggplot2)
```

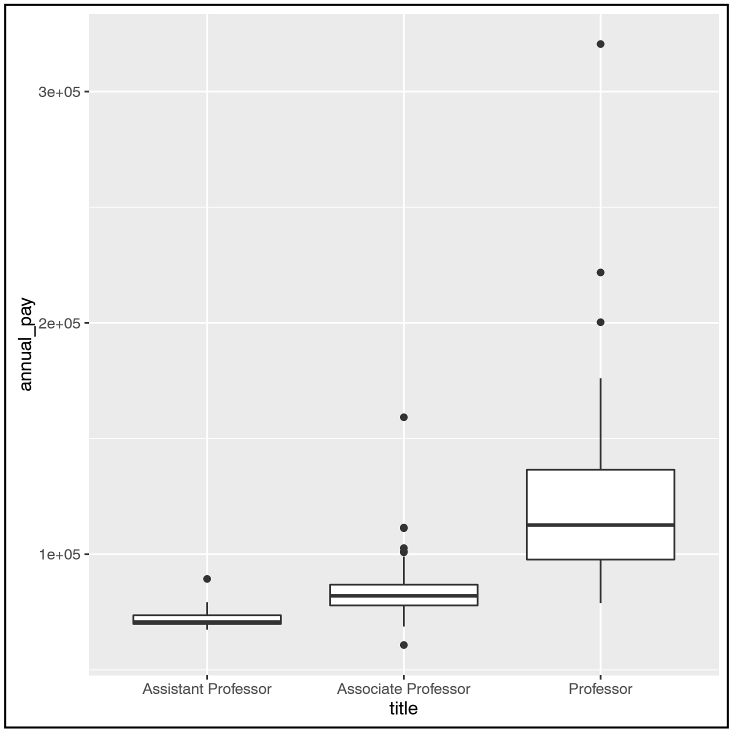
Understanding the Basic Syntax

Aesthetic mappings given in the ggplot() layer are applied to every subsequent layer



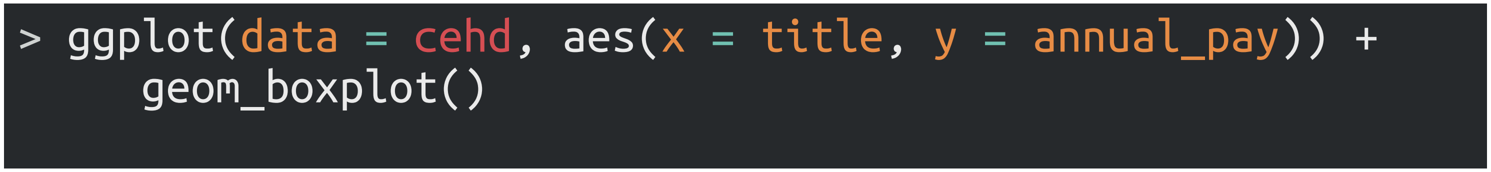
```
> ggplot(data = cehd, aes(x = title, y = annual_pay)) +
    geom_boxplot()
```

The geom_boxplot() function adds the geometric object of boxplots using the global data and aesthetic mapping.



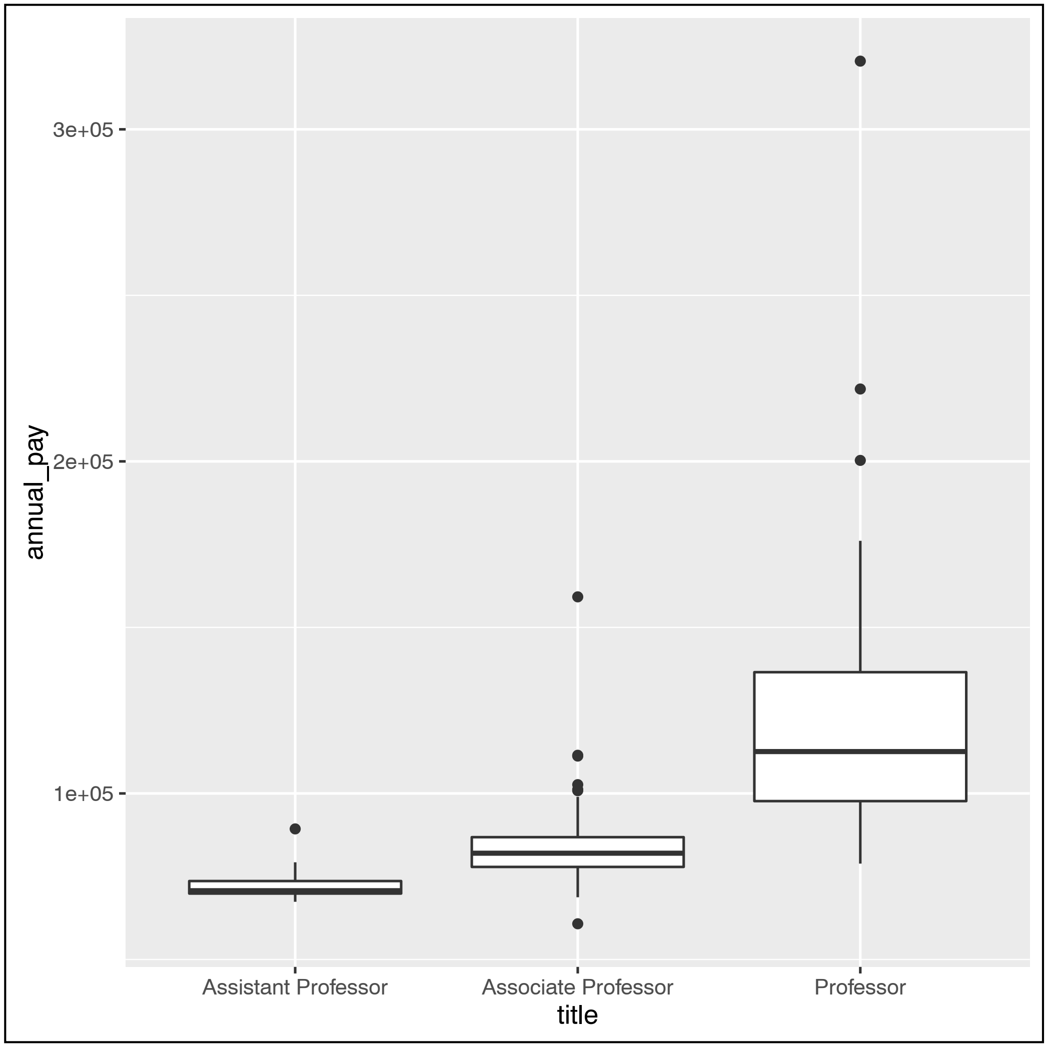
The + adds another layer.

Understanding the Basic Syntax



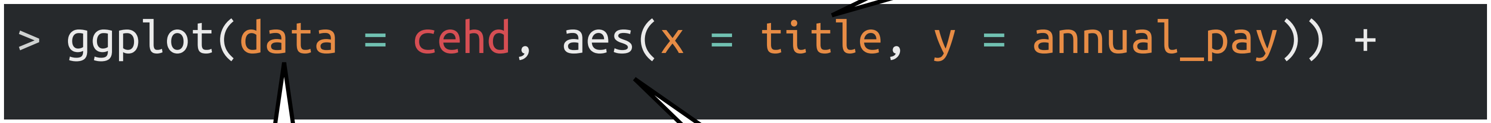
```
> ggplot(data = cehd, aes(x = title, y = annual_pay)) +
    geom_boxplot()
```

Plots are built by layering graphical components. In the syntax, the layers are literally *summed* together to form the plot.



Global Layer

Aesthetic mappings given in the `ggplot()` layer are applied to all layers in the plot



```
> ggplot(data = cehd, aes(x = title, y = annual_pay)) +
```

The `data=` argument indicates the source data frame.

The `aes=` argument sets the aesthetic mapping(s).

The first layer is always `ggplot()`. It contains reference to the **source data** (data frame) and ***global* aesthetic mappings**.

Aesthetic mappings describe how **variables in the data are mapped to visual properties** (aesthetics) of geoms. They are used to define position (x-dimension y-dimension), size, color, fill, groupings, etc.

- Aesthetics can be set **globally**—in `ggplot()` layer—or **locally** (only used in a specific geom layer)
- Each aesthetic can be **variable** or **fixed**
    - If the aesthetic is variable it needs to be specified in the `aes()` function
    - If the aesthetic is fixed it should be specified outside the `aes()` function

Adding Geometric Objects

```
> ggplot(data = cehd, aes(x = title, y = annual_pay)) +
    geom_boxplot()
```

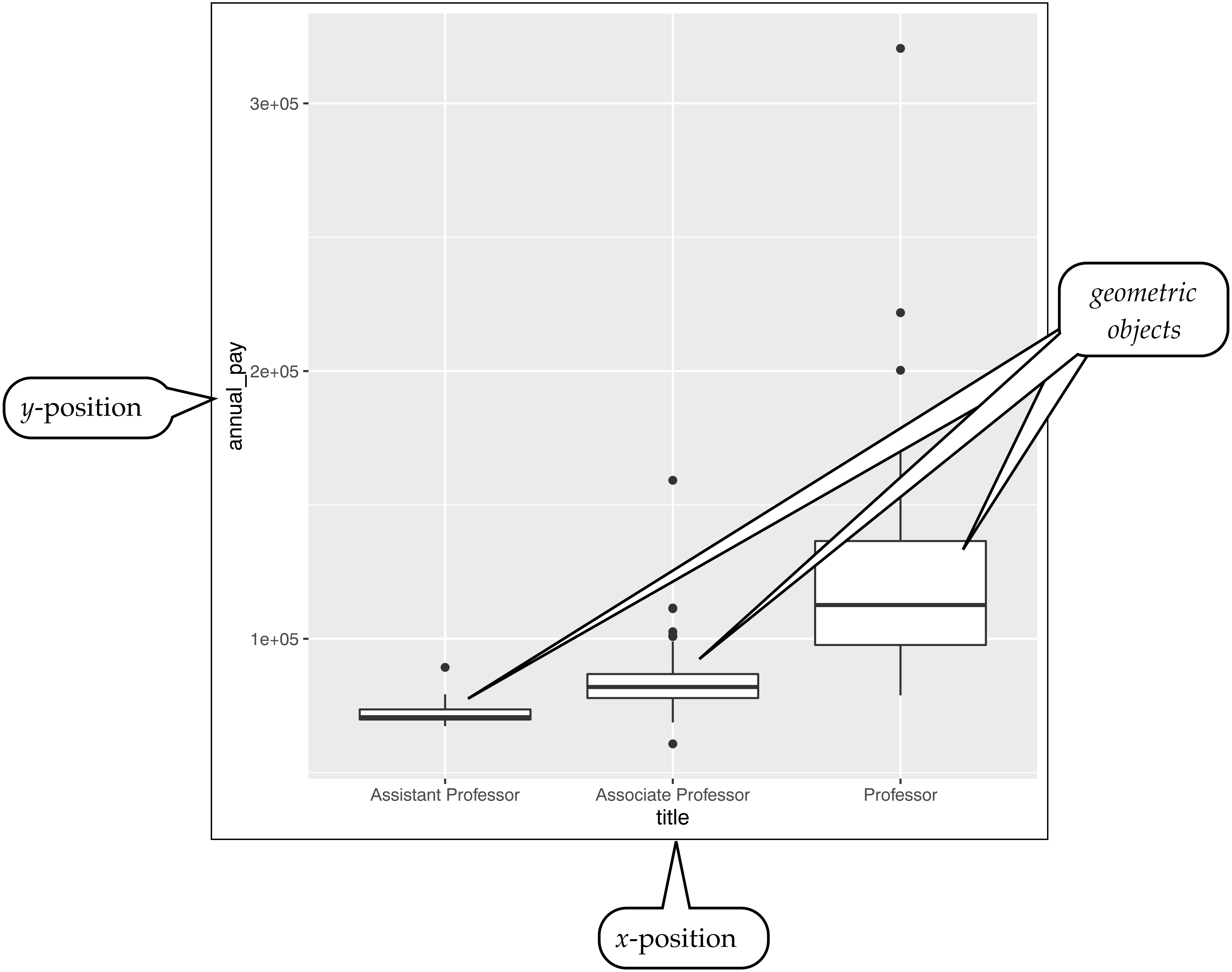
The `geom_boxplot()` function adds the geometric object of boxplots using the global data and aesthetic mapping.

The + adds another layer.

The first layer only sets up the plot, it doesn't actually plot anything. In the subsequent layers, we add geometric objects (e.g., points, boxplots). These objects are plotted based on the aesthetics from the first layer. For example, the syntax above draws boxplots of faculty members' annual pay vertically (the *y*-position is the variable *annual_pay*. The *x*-position is *title*, so each title will have its own box plot, separated along the *x*-axis.

Geometric objects, or *geoms*, are features that are actually drawn on plot (e.g., lines, points). They are specified using the prefix `geom_` and a suffix that names the feature to be plotted.

- **Points** specified with `geom_point()`
- **Jittered points** specified with `geom_jitter()`
- **Lines** specified with `geom_line()`
- **Boxplots** specified with `geom_boxplot()`

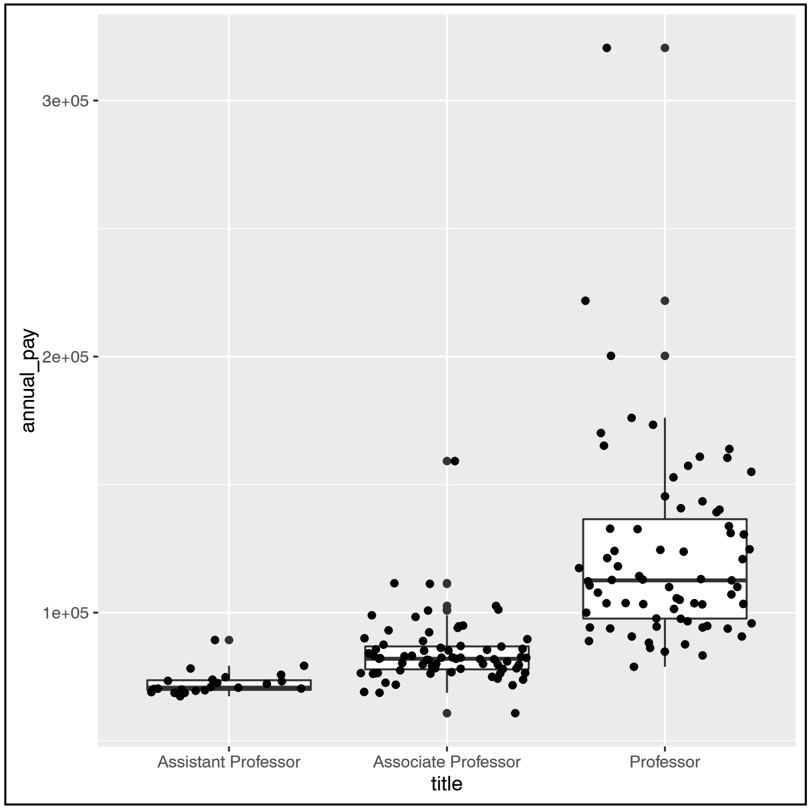

3e+05
2e+05
1e+05
annual_pay
Assistant Professor
Associate Professor
Professor
title
y-position
x-position
geometric objects

When layers are added they are "stacked" on top of previous layers. Imagine drawings on separate transparencies, and then those transparencies are stacked.

```
> ggplot(data = cehd, aes(x = title, y = annual_pay)) +
    geom_boxplot() +
    geom_jitter()
```

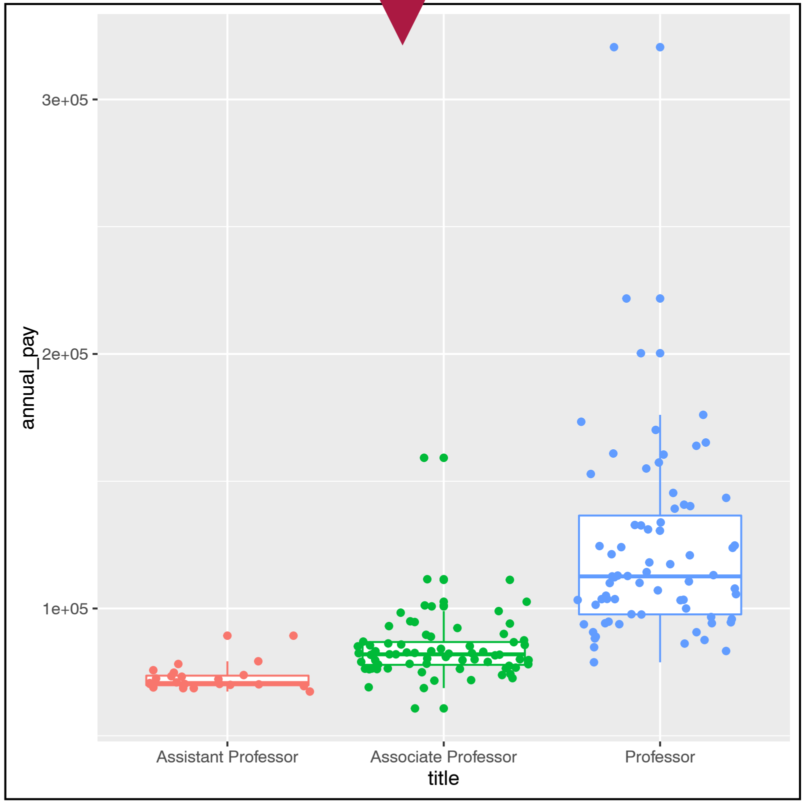
The + adds another layer.

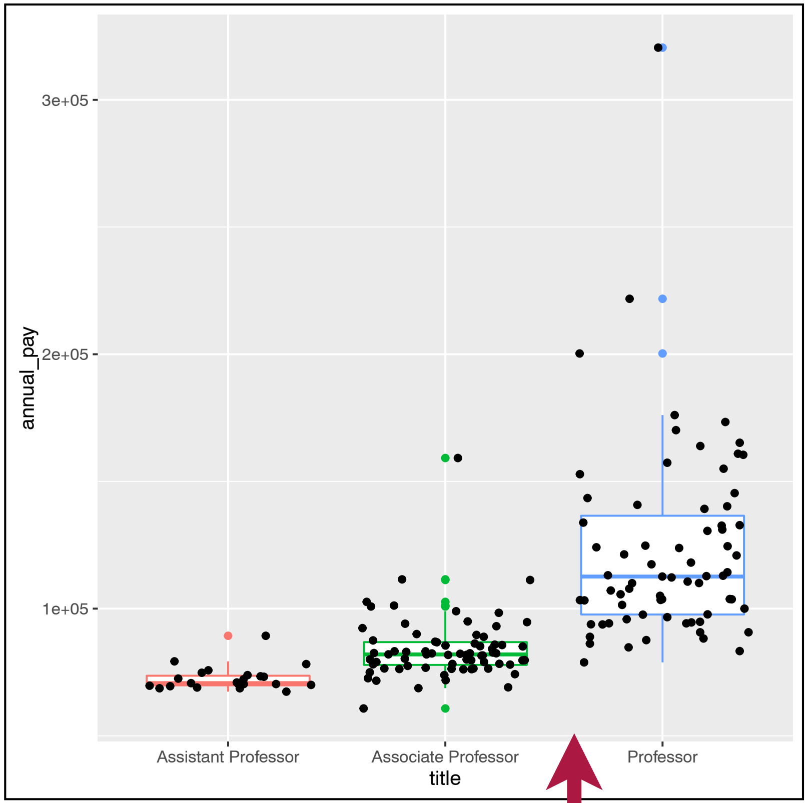
The geom_jitter() function adds the geometric object of jittered points using the global data and aesthetic mapping.



```
> ggplot(data = cehd, aes(x = title, y = annual_pay, color = title)) +
    geom_boxplot() +
    geom_jitter()
```

Global aesthetic mappings are applied to *all* layers.





```
> ggplot(data = cehd, aes(x = title, y = annual_pay)) +
    geom_boxplot(aes(color = title)) +
    geom_jitter()
```

Local aesthetic mappings (in a particular layer) are only applied to that layer.

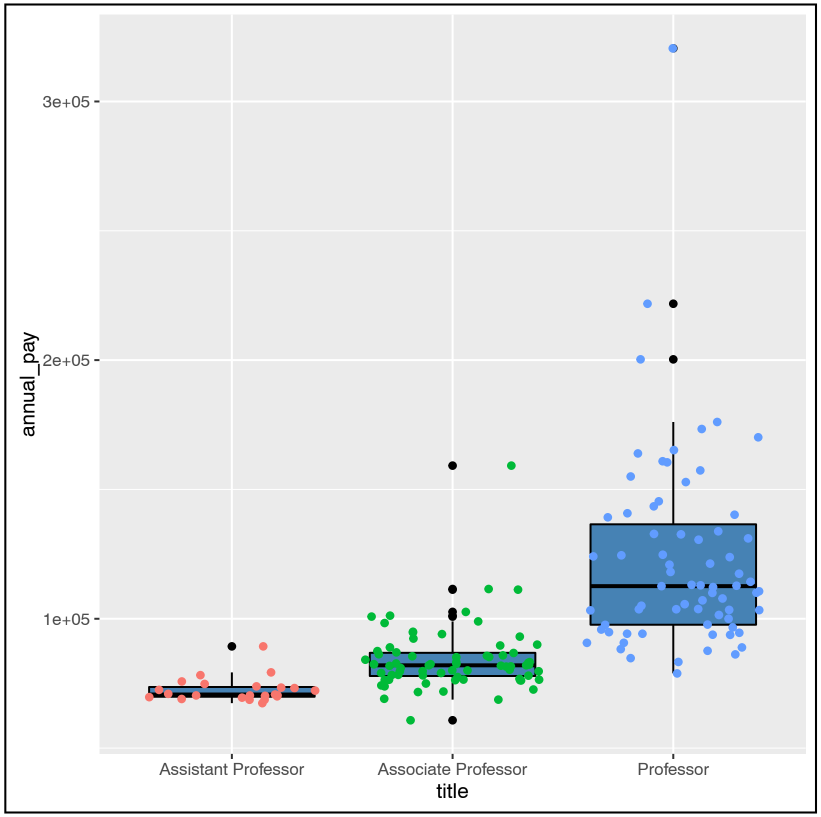
Fixed vs Variable Aesthetics

```
> ggplot(data = cehd, aes(x = title, y = annual_pay, color = title)) +
    geom_boxplot(color = "black", fill = "steelblue") +
    geom_jitter()
```

The `color=` argument sets the color for the outline in this layer.

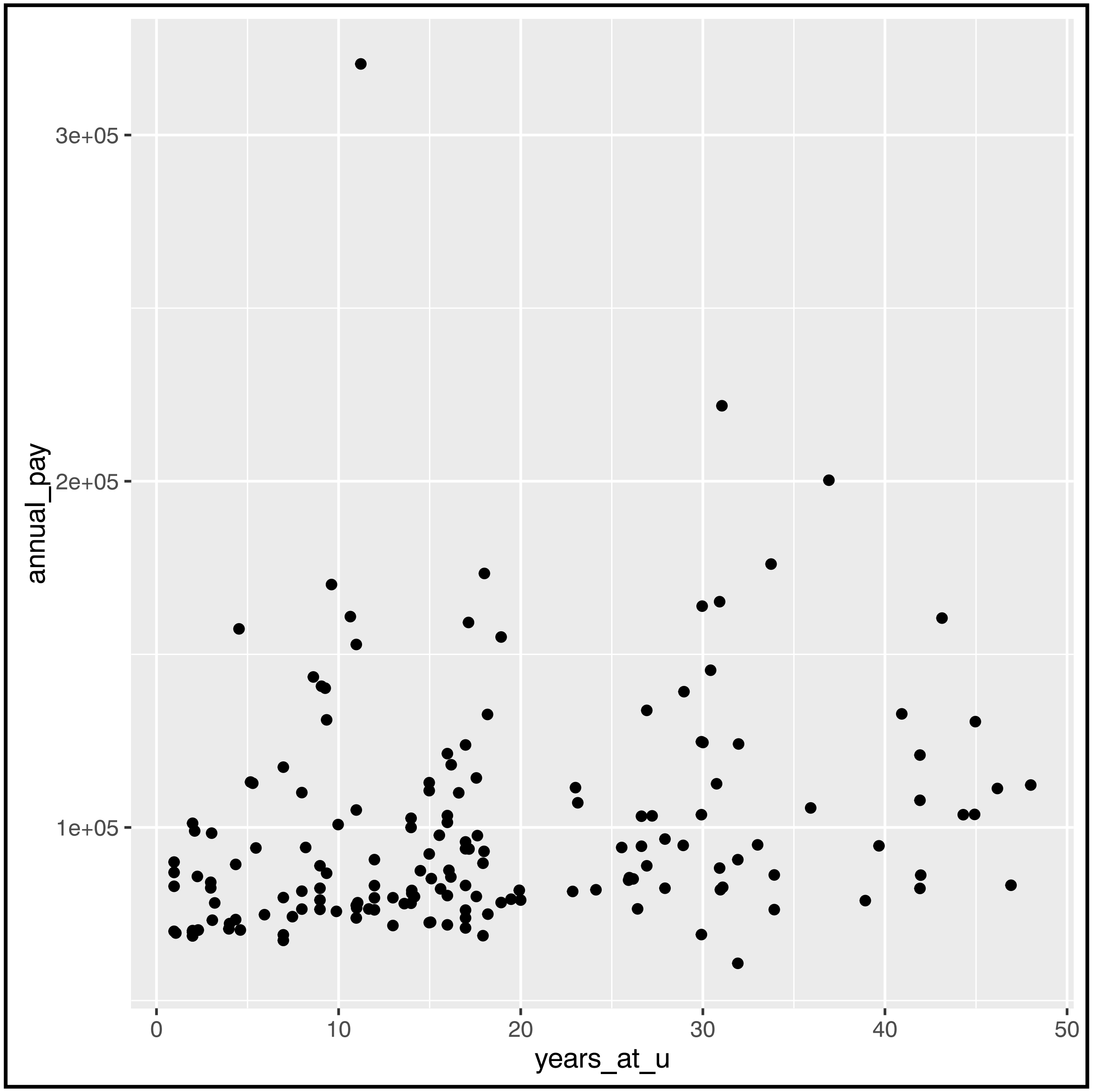
The `fill=` argument sets the fill color for this layer.

Notice the quotation marks...color names are character strings.



Aesthetic mappings that are fixed to a particular value (do not vary), rather, do **not** need to be enclosed in the `aes()` function.

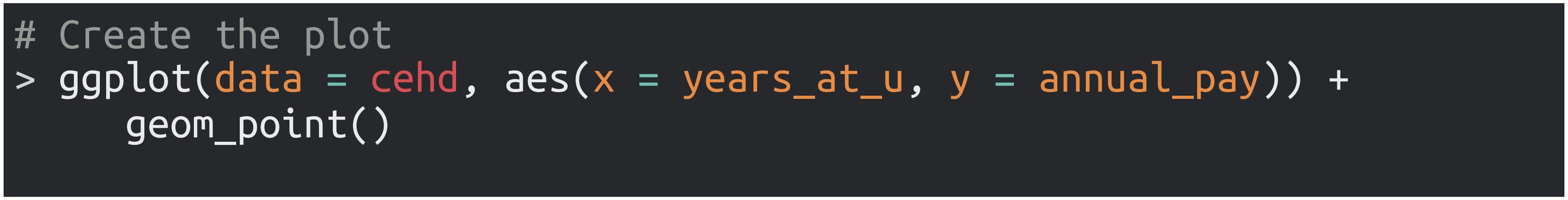
Note also that the local aesthetics override the global aesthetics



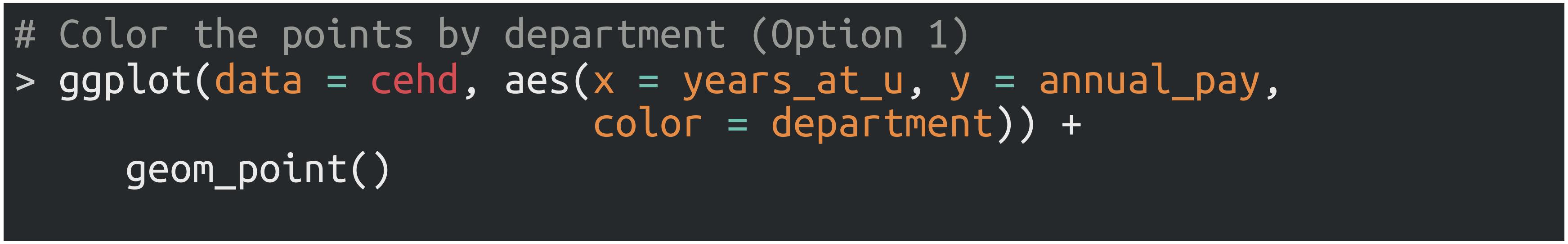
Your Turn

Write the syntax to create this scatterplot. Hint to add points, we use `geom_point()`

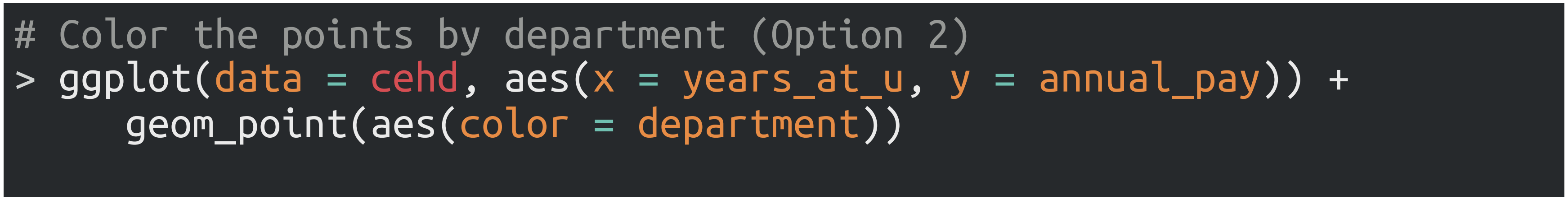
How would we color the points by department?



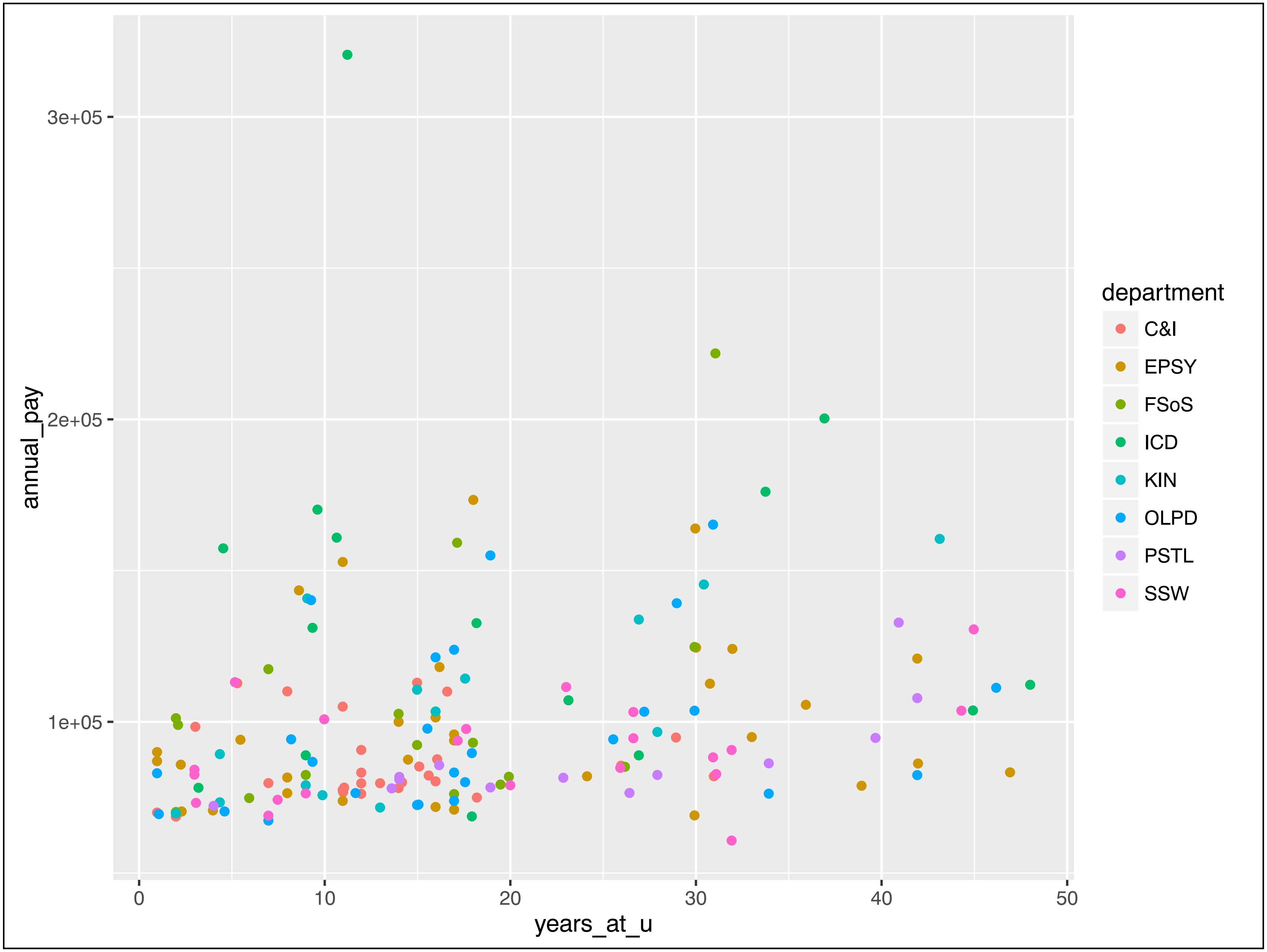
```r
# Create the plot
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point()
```



```r
# Color the points by department (Option 1)
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay,
                          color = department)) +
     geom_point()
```



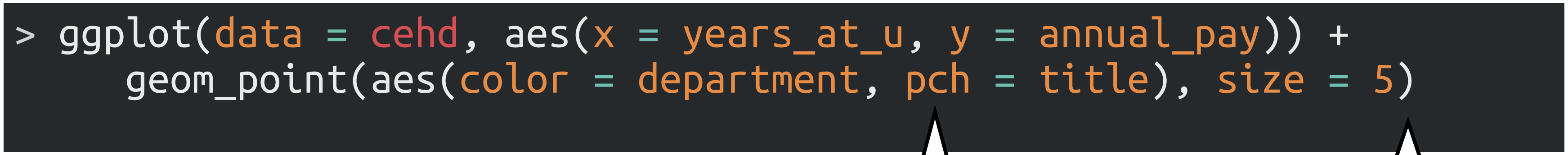
```r
# Color the points by department (Option 2)
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point(aes(color = department))
```



When we use non-positional aesthetics (e.g., color) ggplot will add a legend to our plot.

Point Aesthetics

Two other useful aesthetics for points are pch= and size= for plotting character and point size, respectively.

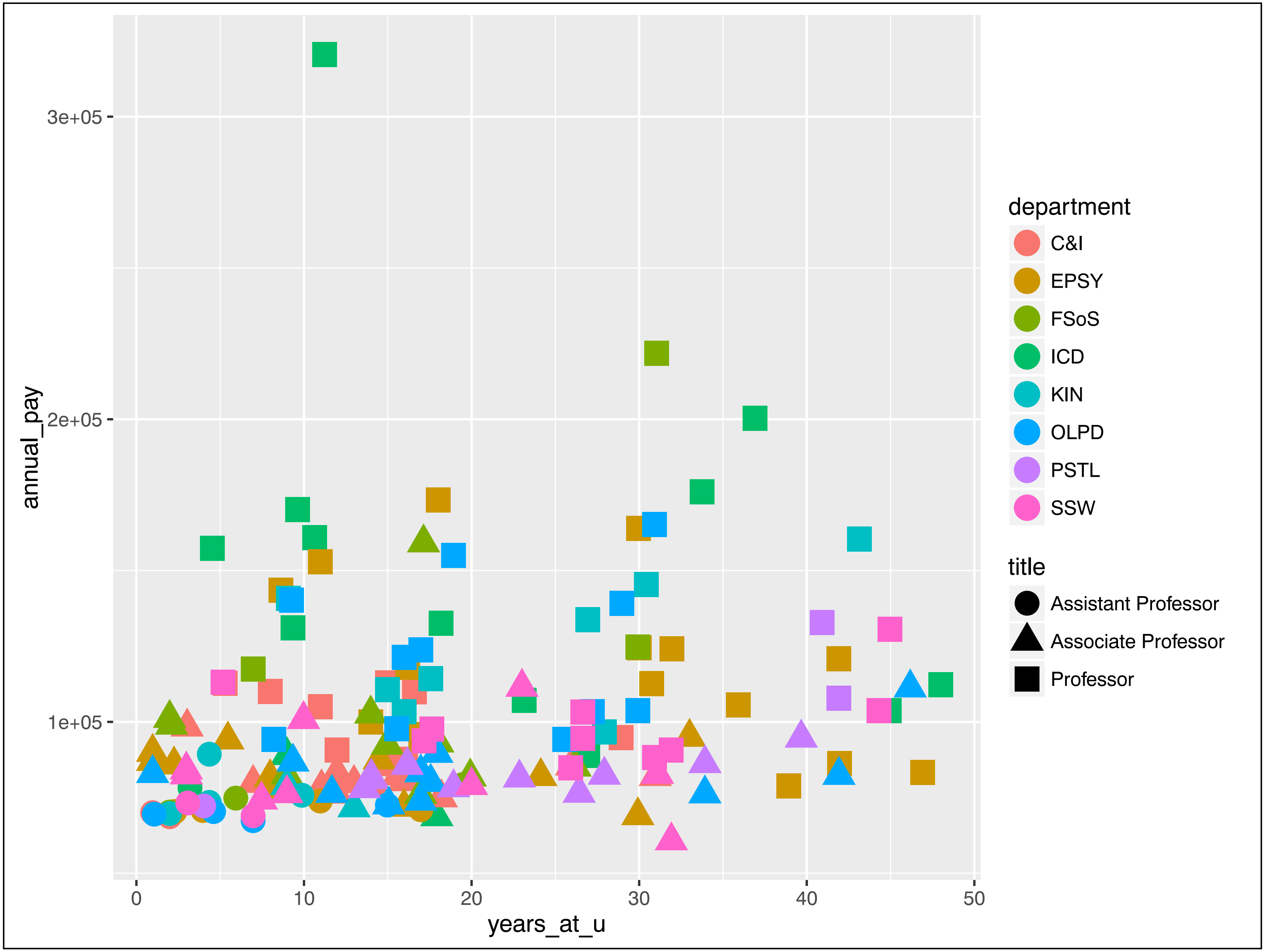


```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = department, pch = title), size = 5)
```

The pch= argument sets the plotting character.

The size= argument sets the point size. (The default size is 4.)

Describe the resulting plot based on the syntax above.



Note: EVERY non-positional aesthetics gets added to the legend.

Faceting

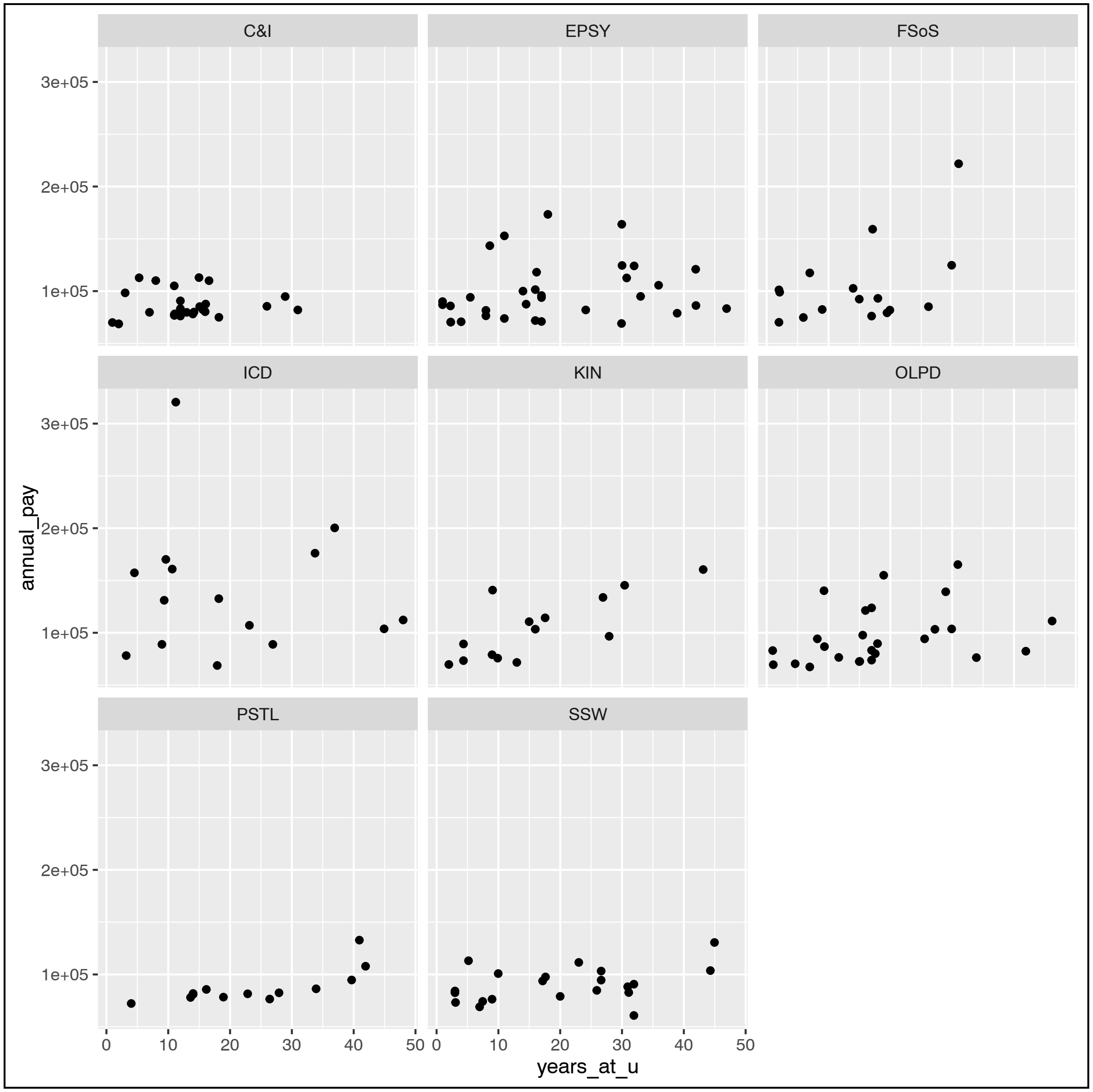
Faceting creates a separate plot for each subgroup declared

- `facet_wrap()` displays the plots conditioned on a **single predictor**
- `facet_grid()` displays the plots conditioned on **multiple predictors**

```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point() +
     facet_wrap(~ department)
```

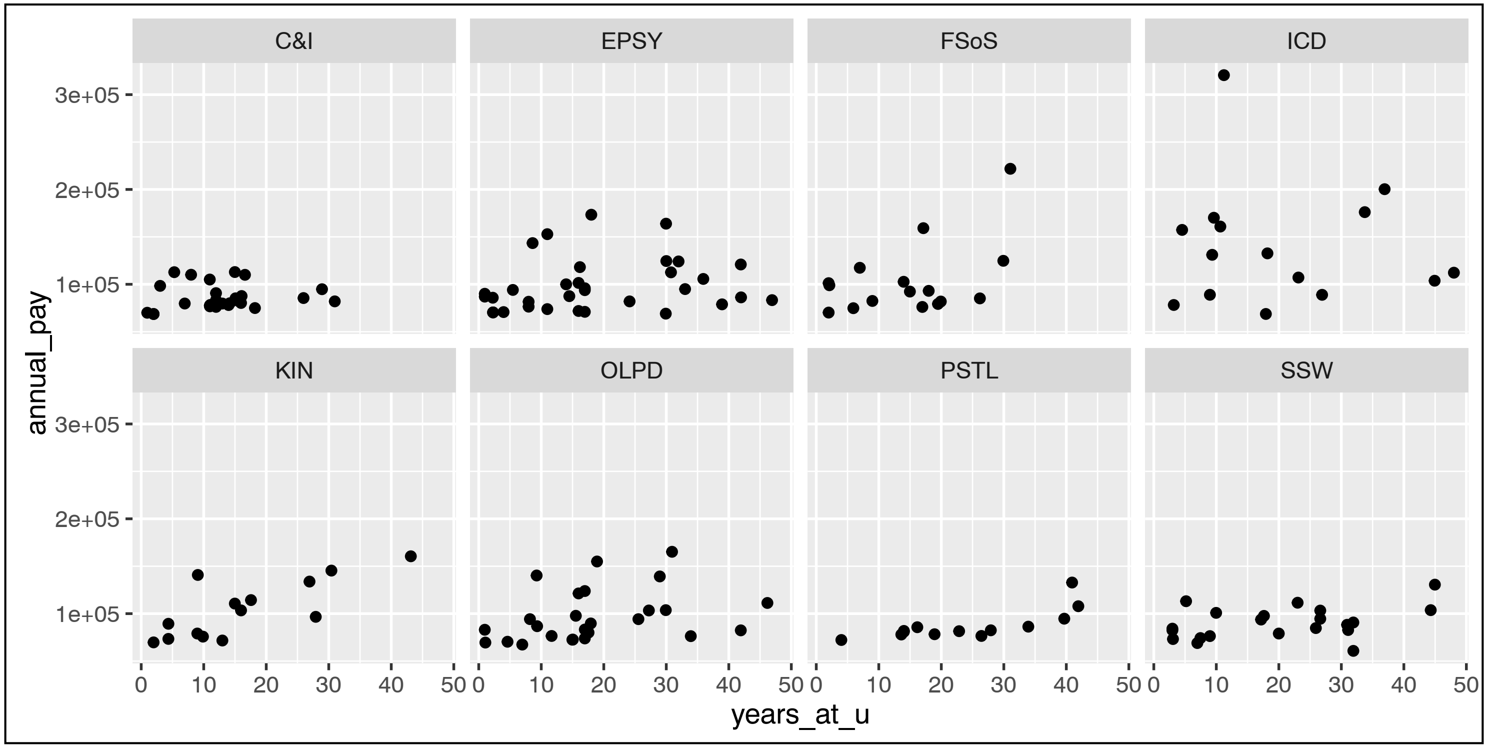
~ sets the predictor for conditioning

The scatterplots show the relationship between experience (years at the university) and pay conditioned on department.

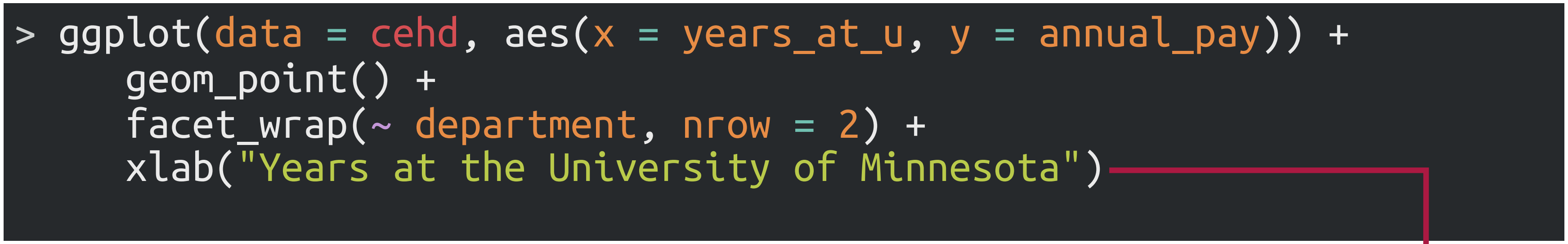


```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point() +
    facet_wrap(~ department, nrow = 2)
```

`nrow=` (and/or `ncol=`) sets the number of rows or columns in the plotting area

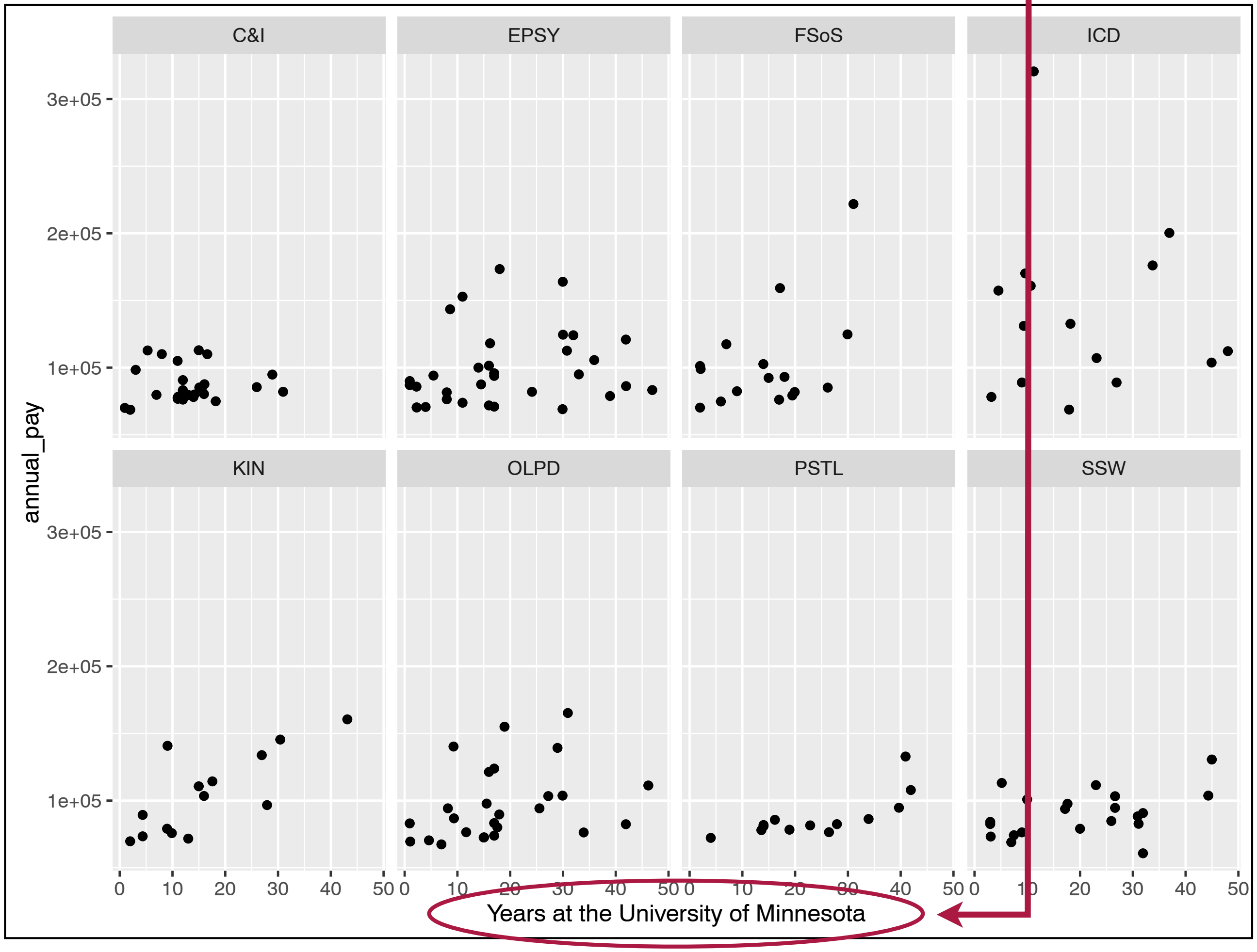


Changing the Axis Label



```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point() +
     facet_wrap(~ department, nrow = 2) +
     xlab("Years at the University of Minnesota")
```

`xlab()` can be used to change the label on the *x*-axis, and `ylab()` is used to change the label on the *y*-axis.

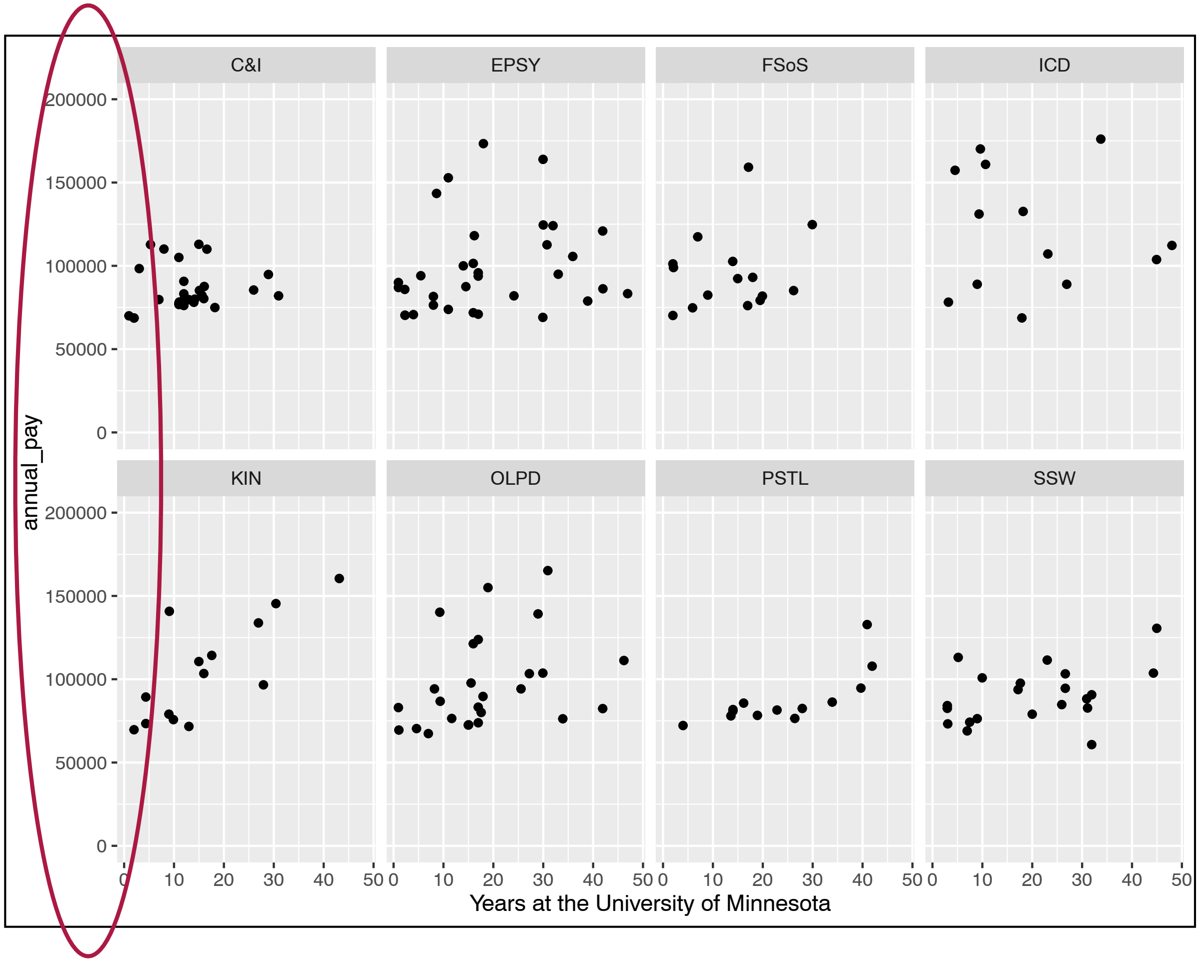


Changing the Axis Limits

```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point() +
    facet_wrap(~ department, nrow = 2) +
    xlab("Years at the University of Minnesota") +
    ylim(0, 200000)
```

The first value is the minimum.

The second value is the maximum.



`xlim()` and `ylim()` are used to set the limits on the *x*-axis and *y*-axis respectively.

Adios-ing Scientific Notation

Because of the high salaries of a few faculty members, R labelled the values using scientific notation. We can force R to not use scientific notation by setting R's global options with the `options()` function.

```
> options(scipen = 10000)

> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point() +
    facet_wrap(~ department, nrow = 2) +
    xlab("Years at the University of Minnesota")
```

The `scipen=` option takes a numeric value that helps R decide whether to print numeric values in fixed or exponential notation. Positive values bias towards fixed and negative towards scientific notation: fixed notation will be preferred unless it is more than *scipen* digits wider..

Fine-Tuning Axis Scales

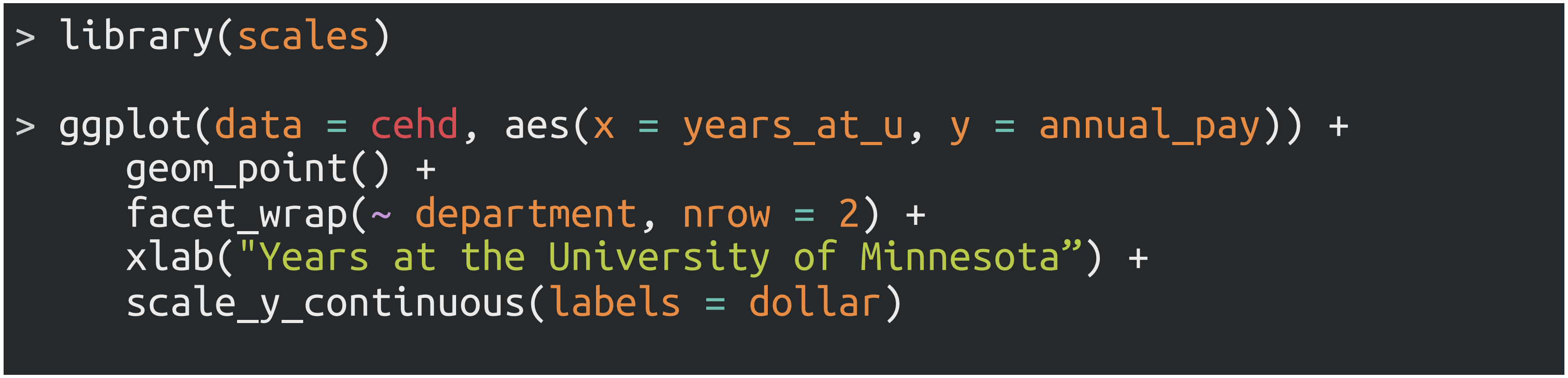
Adding a `scale()` layer allows much more fine-tuning of the axis scales. We have to specify which axis (*x* or *y*) and whether the variable plotted along that axis is *continuous* or *discrete*. To fine-tune the *y*-axis in our example, we would use the `scale_y_continuous()` layer.

```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point() +
     facet_wrap(~ department, nrow = 2) +
     xlab("Years at the University of Minnesota") +
     scale_y_continuous(
         name = "Years at the University of Minnesota”,
         breaks = c(50000, 100000, 150000, 200000, 250000, 300000)
     )
```

The `name=` option labels the scale (it is the same as the `ylab()` layer in this case). The `breaks=` option adds break lines on the axis. There are several other options including `labels=` for labelling the break lines, etc.

Prettying Up the Scales

We can get other options for labeling using the **scales** package. For example, we can add commas to separate by thousands in long values, or add the $ for monetary values.



```
> library(scales)

> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point() +
    facet_wrap(~ department, nrow = 2) +
    xlab("Years at the University of Minnesota”) +
    scale_y_continuous(labels = dollar)
```

The `labels=dollar` option is a built-in formatter from the **scales** package that adds the dollar sign and commas to the labels on a specified axis. Read more at http://www.rdocumentation.org/packages/scales/versions/0.4.0

Customizing the Color

scale() functions can also be used to fine-tune colors and fills. For these you need to specify either *color* or *fill*, and also the palette you want to use. For example, scale_color_manual() can be used to manually set the colors when the color= argument is used.

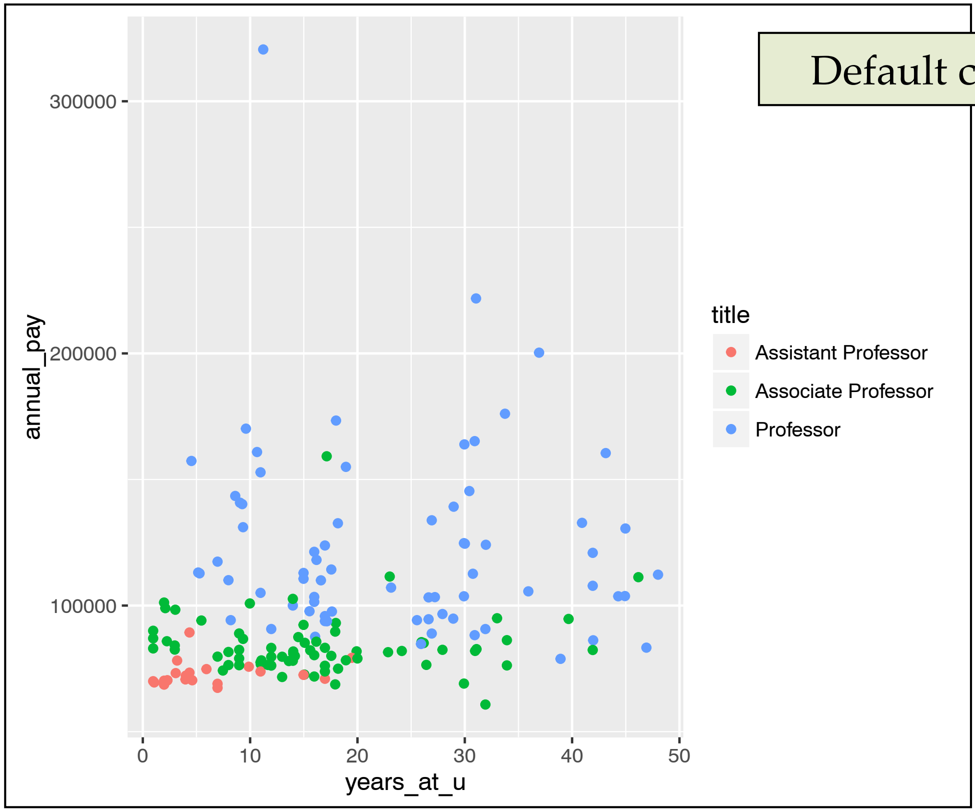
```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = title)) +
    scale_fill_manual(
        values = c("#599ad3", "#f9a65a", "#9e66ab")
        )
```

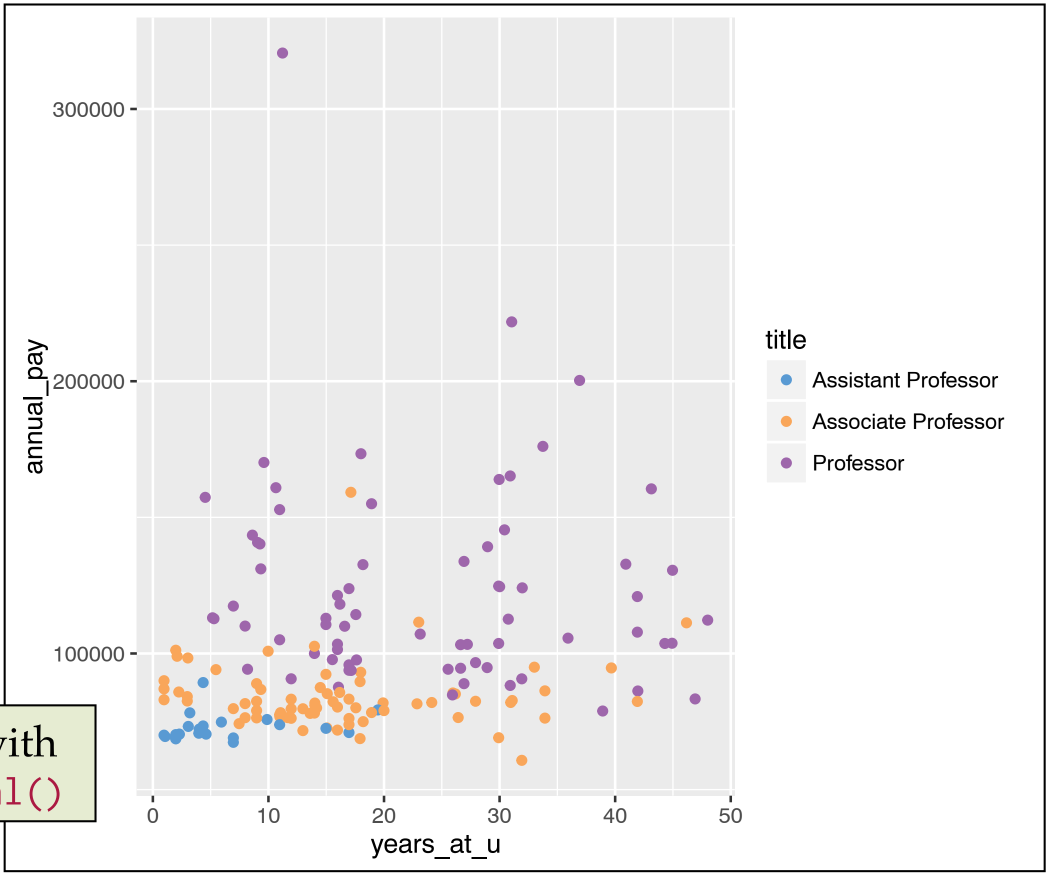
scale_fill_manual() allows you to manually set the attributes associated with the fill aesthetic.

The values= argument sets the color values for each level of the factor.

Named colors or HEX values (both given as quoted character strings) can be used in values= argument of scale_color_manual() or scale_fill_manual().

Default color palette



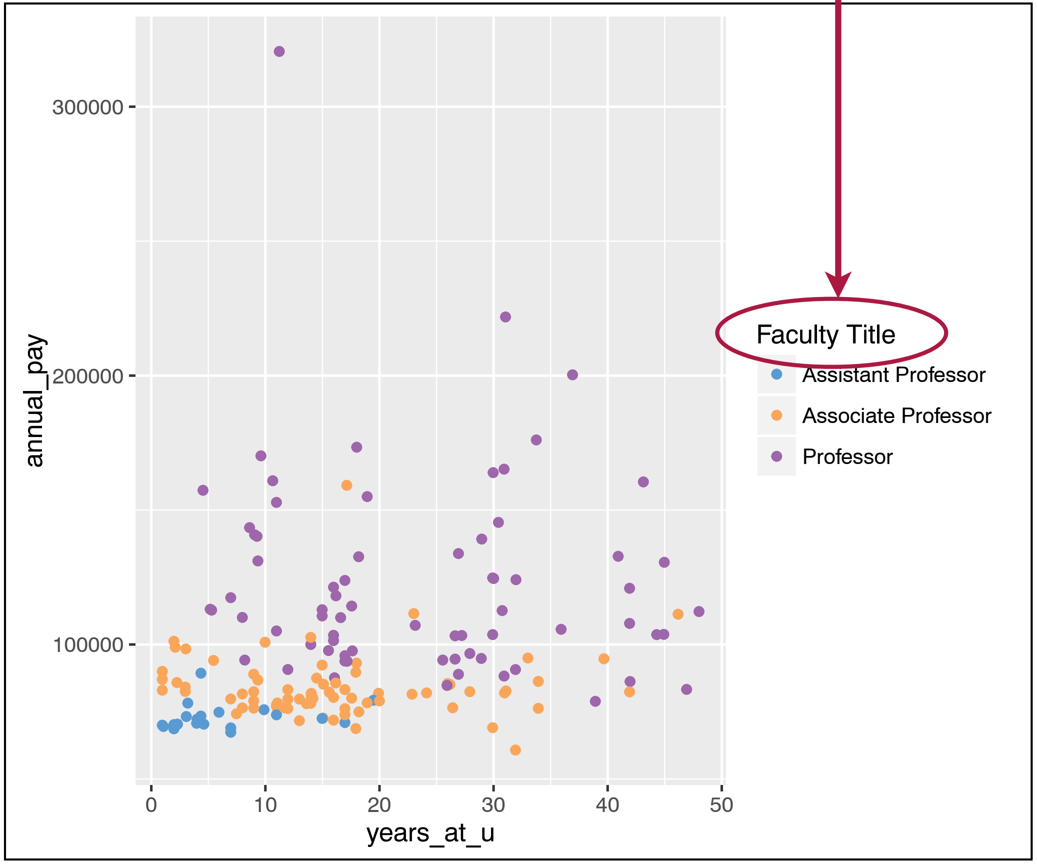


Palette provided with `scale_color_manual()`

scale() functions can also be used to change the name and labels in the legend.

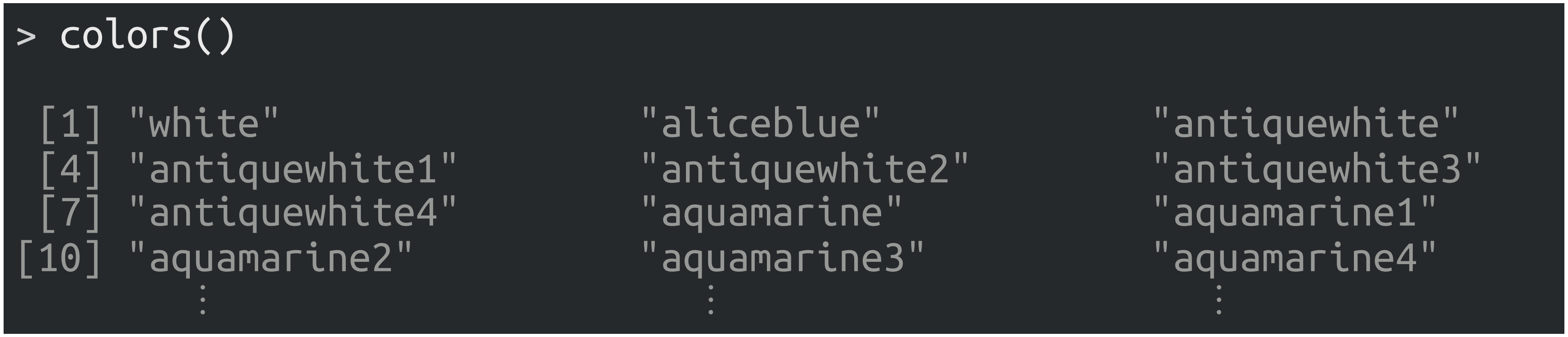
```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = title)) +
    scale_fill_manual(
         values = c("#599ad3", "#f9a65a", "#9e66ab"),
         name = "Faculty Title"
         )
```

The name= argument changes the title of the legend.



Choosing a Color Palette

`colors()` will provide a list of all the **named colors** available in R.

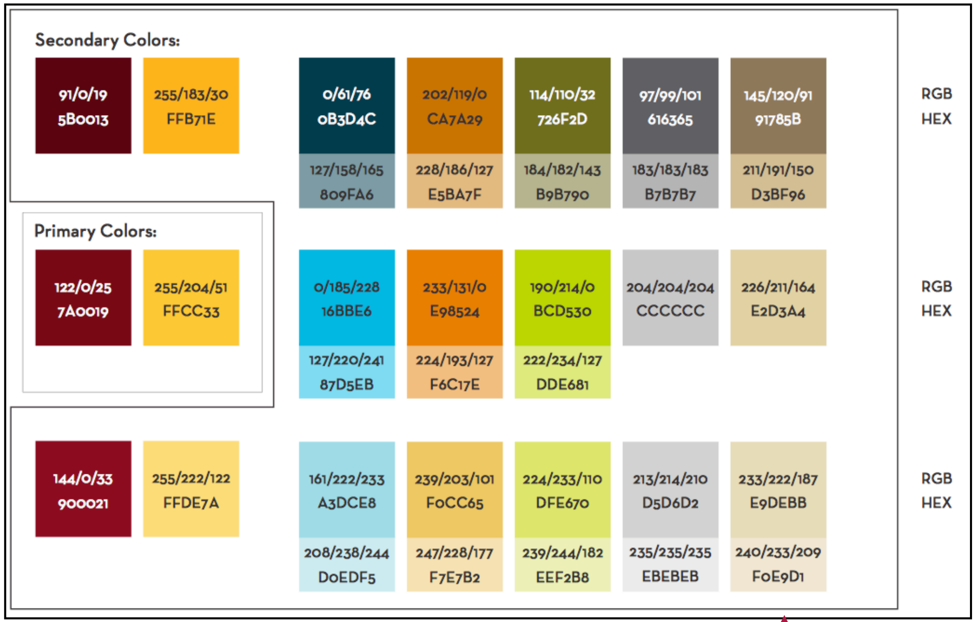


```
> colors()

  [1] "white"                "aliceblue"            "antiquewhite"
  [4] "antiquewhite1"        "antiquewhite2"        "antiquewhite3"
  [7] "antiquewhite4"        "aquamarine"           "aquamarine1"
 [10] "aquamarine2"          "aquamarine3"          "aquamarine4"
        ⋮                      ⋮                      ⋮
```

Most universities have official colors. The University of Minnesota's two official colors in HEX (for electronic display) are:

- `#ffcc33` (gold)
- `#7a0019` (maroon)

See more at: https://www.ur.umn.edu/brand/requirements-and-guidelines/color-and-type/



Secondary Colors:

91/0/19 5B0013 | 255/183/30 FFB71E | 0/61/76 0B3D4C | 202/119/0 CA7A29 | 114/110/32 726F2D | 97/99/101 616365 | 145/120/91 91785B | RGB HEX

127/158/165 809FA6 | 228/186/127 E5BA7F | 184/182/143 B9B790 | 183/183/183 B7B7B7 | 211/191/150 D3BF96

Primary Colors:

122/0/25 7A0019 | 255/204/51 FFCC33 | 0/185/228 16BBE6 | 233/131/0 E98524 | 190/214/0 BCD530 | 204/204/204 CCCCCC | 226/211/164 E2D3A4 | RGB HEX

127/220/241 87D5EB | 224/193/127 F6C17E | 222/234/127 DDE681

144/0/33 900021 | 255/222/122 FFDE7A | 161/222/233 A3DCE8 | 239/203/101 F0CC65 | 224/233/110 DFE670 | 213/214/210 D5D6D2 | 233/222/187 E9DEBB | RGB HEX

208/238/244 D0EDF5 | 247/228/177 F7E7B2 | 239/244/182 EEF2B8 | 235/235/235 EBEBEB | 240/233/209 F0E9D1

The U of M also has an entire palette of secondary colors available at: https://www.ur.umn.edu/brand/assets/pdf/secondary_colors_rgb.pdf

Pre-Selected Color Palettes

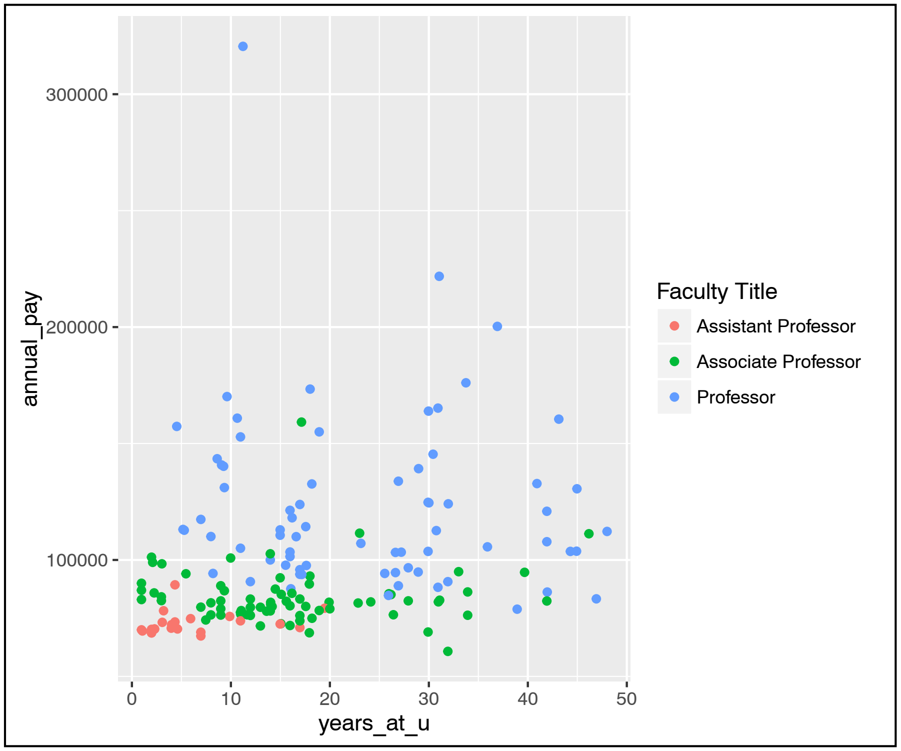
There are several "built-in" color palettes available for use in ggplot

Fill Scale	Color Scale	Description
`scale_fill_hue()`	`scale_color_hue()`	Colors evenly spaced around the color wheel
`scale_fill_grey()`	`scale_color_grey()`	Grey scale palette
`scale_fill_brewer()`	`scale_color_brewer()`	ColorBrewer palettes

Default Color Palette

The `scale_color_hue()` and `scale_fill_hue()` functions use the default color palette. They are useful for changing the name and labels in the legend if you want to use the default palette.

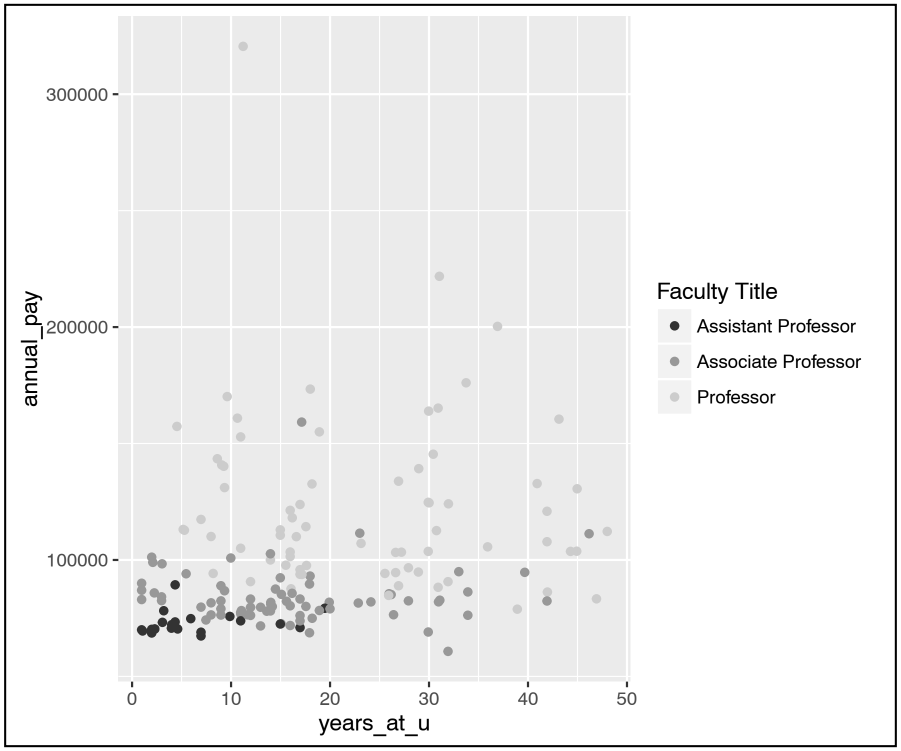
```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = title)) +
    scale_color_hue(name = “Faculty Title”)
```



Grey Scale Color Palette

The `scale_color_grey()` and `scale_fill_grey()` functions use a greyscale color palette. This is a useful palette if you are printing in black-and-white.

```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point(aes(color = title)) +
     scale_color_grey(name = “Faculty Title”)
```



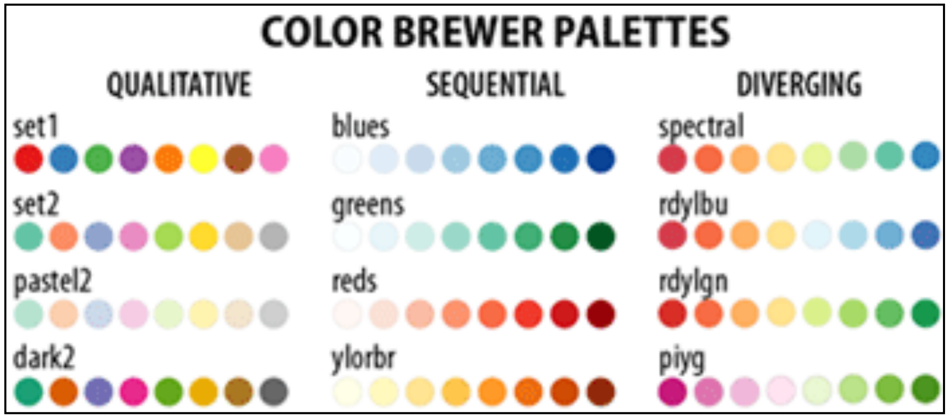
Color Brewer

Cynthia Brewer chose color palettes that not only are aesthetically pleasing, but also based on how humans perceive the colors that are displayed.

http://www.colorbrewer2.org

She has palettes for three different types of data

- **Qualitative/Categorical**—colors do not have a perceived order
- **Sequential**—colors have a *perceived order* and perceived difference between successive colors is uniform
- **Diverging—**two back-to-back sequential palettes starting from a common color (e.g., for Likert scale data)

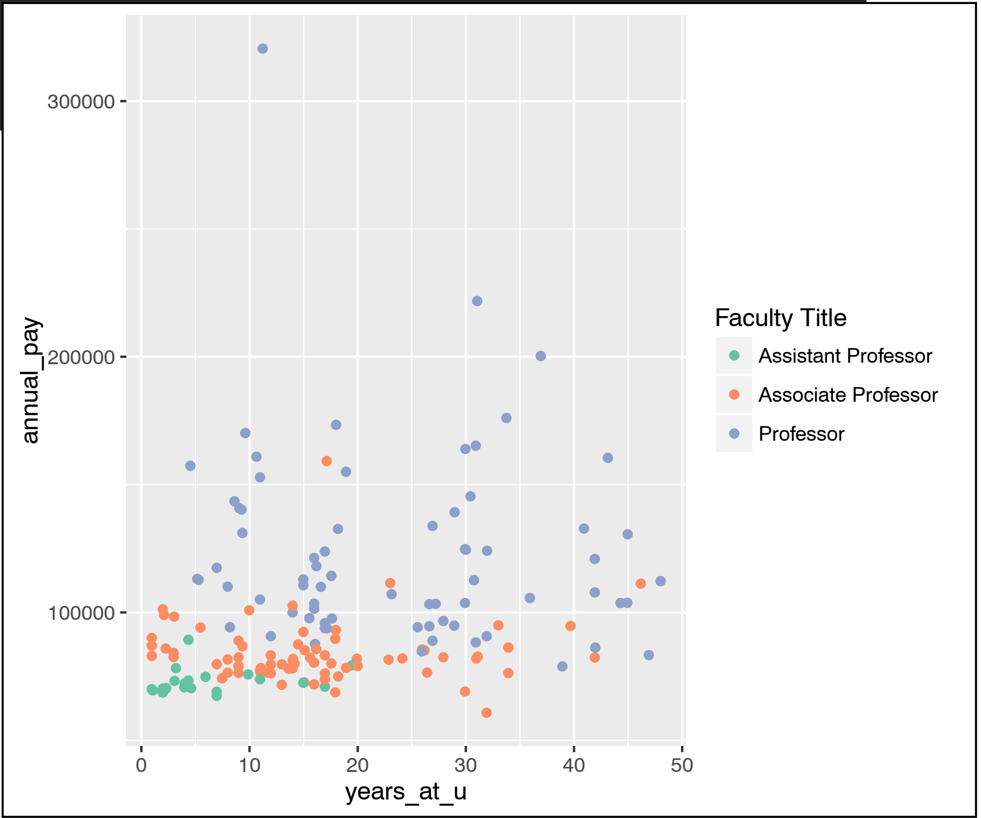


There is a very readable introduction to color brewer palettes at http://mkweb.bcgsc.ca/brewer/

Brewer Color Palette

The `scale_color_brewer()` and `scale_fill_brewer()` functions use a Cynthia Brewer's color palettes. You need to specify a palette using the `palette=` argument.

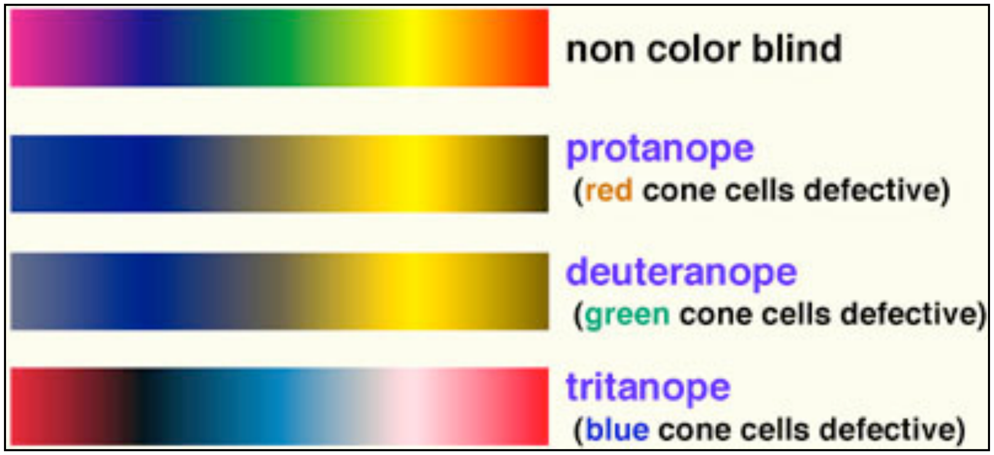
```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = title)) +
    scale_color_brewer(
        name = "Faculty Title",
        palette = "Set2"
        )
```



Palettes for Color-Blindness

About 8% of males and ½% of females have some form of color vision deficiency (good chance that someone in your audience will be one of these people)

Color *and* grey-scale palettes have been developed for use with people that have the more common forms of color-blindness



There is more information related to color-blindness and the creation of suitable color palettes for scientific figures at http://jfly.iam.u-tokyo.ac.jp/color/

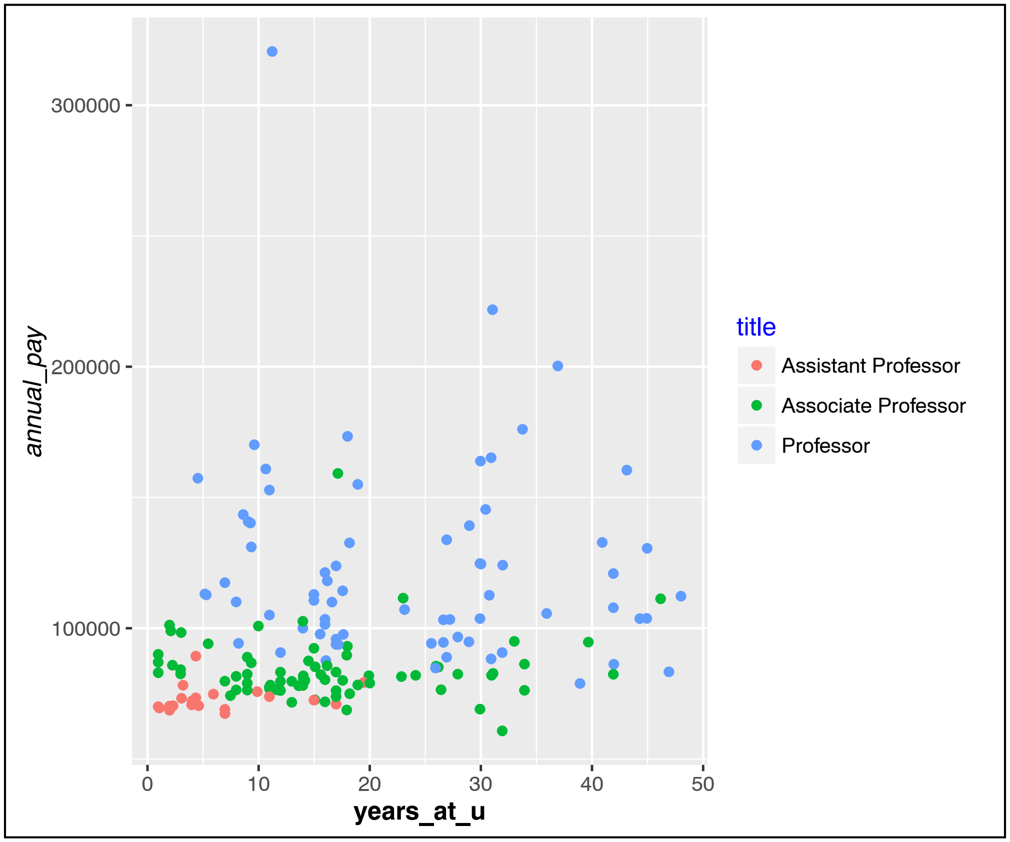
There is a large body of research literature related to the creation of suitable color palettes for figures. As a starting point,

Lumley, T. (2006). Color-coding and color blindness in statistical graphics. *Statistical computing and graphics newsletter*. http://www.amstat-online.org/sections/graphics/newsletter/Volumes/v172.pdf

Fine-Tuning the Theme

```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point(aes(color = department)) +
     theme(
          axis.title.x = element_text(face = "bold"),
          axis.title.y = element_text(face = "italic"),
          legend.title = element_text(color = "blue")
          )
```

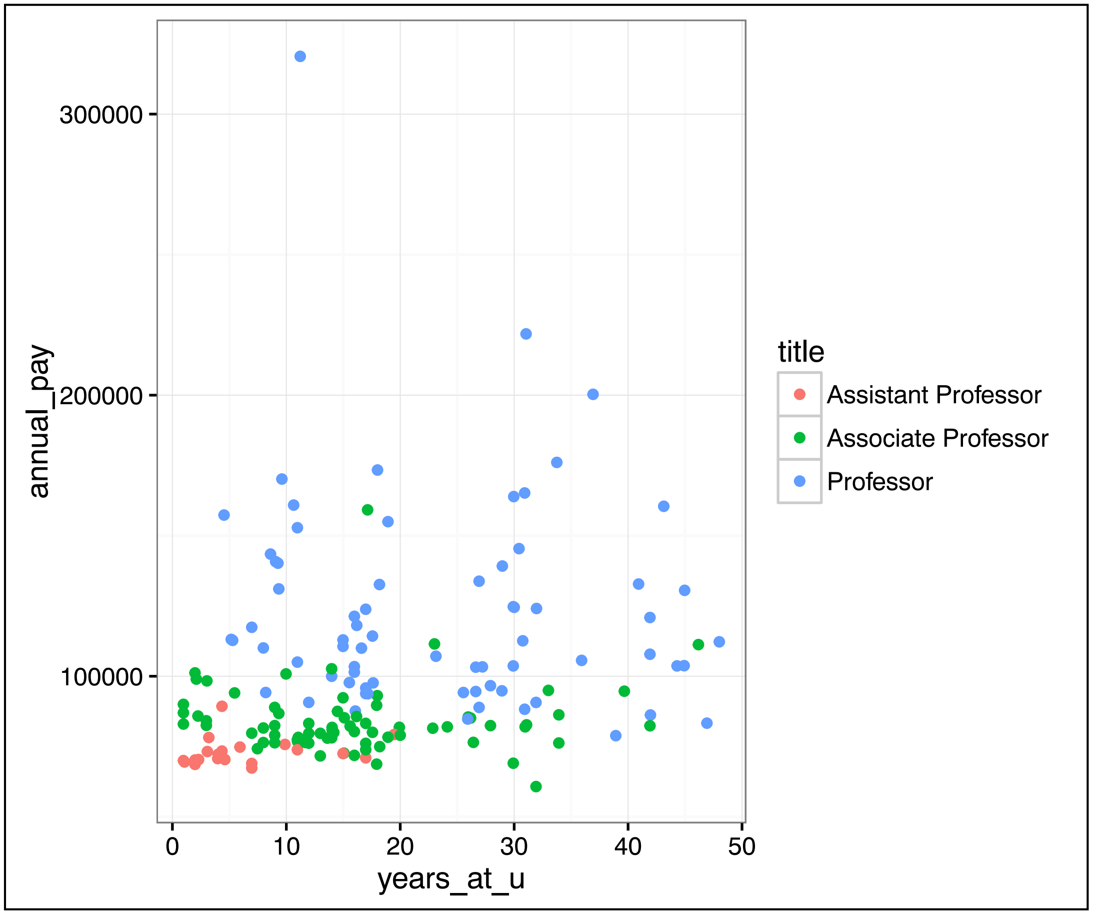
The `theme()` function can be used to change *every* element in the plot (e.g., grid lines, font, color, etc.). See http://docs.ggplot2.org/current/theme.html



Using "Built-In" Themes

```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = department)) +
    theme_bw()
```

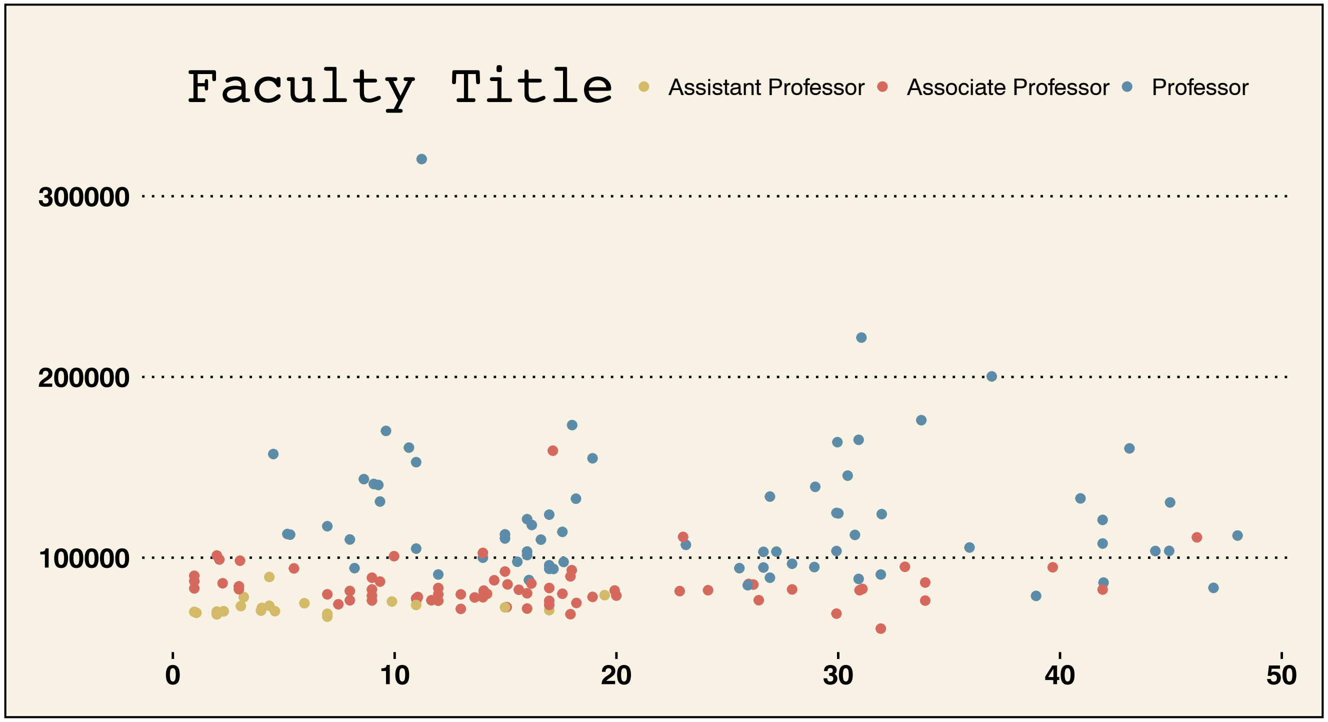
The theme_bw() function is a "built-in" theme that uses a black-and-white background (rather than grey).



There are many other themes available

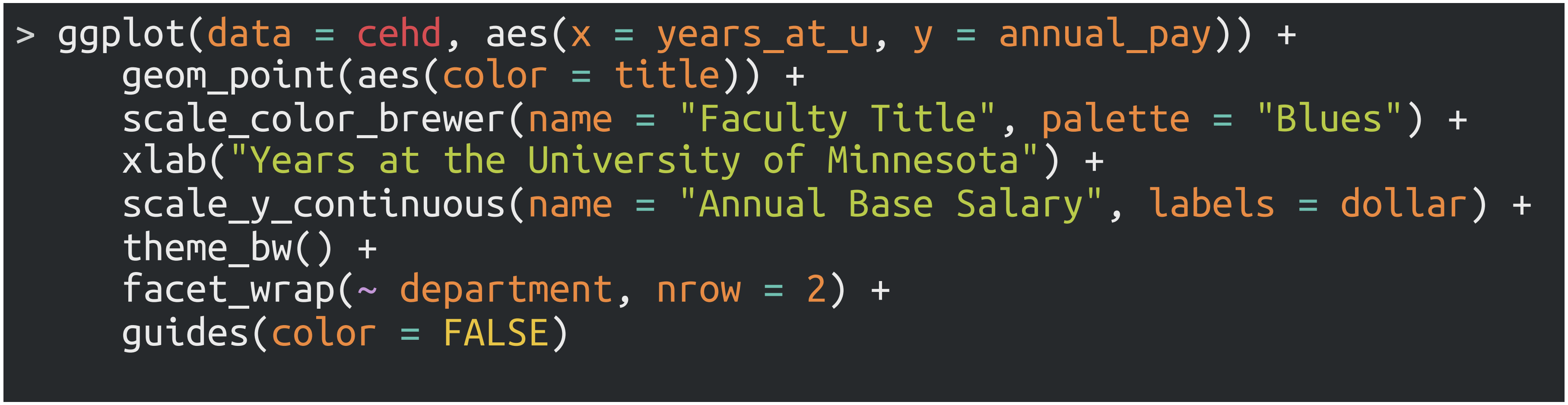
- http://drunks-and-lampposts.com/2012/10/02/clegg-vs-pleb-an-xkcd-esque-chart/
- https://github.com/jrnold/ggthemes

```
# Install the ggthemes library then load it
> library(ggthemes)

> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
     geom_point(aes(color = department)) +
     theme_wsj() +
     scale_color_wsj(name = "Faculty Title", palette = "rgby")
```



You can also build your own themes and use them.

Putting It All Together



```
> ggplot(data = cehd, aes(x = years_at_u, y = annual_pay)) +
    geom_point(aes(color = title)) +
    scale_color_brewer(name = "Faculty Title", palette = "Blues") +
    xlab("Years at the University of Minnesota") +
    scale_y_continuous(name = "Annual Base Salary", labels = dollar) +
    theme_bw() +
    facet_wrap(~ department, nrow = 2) +
    guides(color = FALSE)
```

Make a rough sketch of the plot you think this syntax will produce.

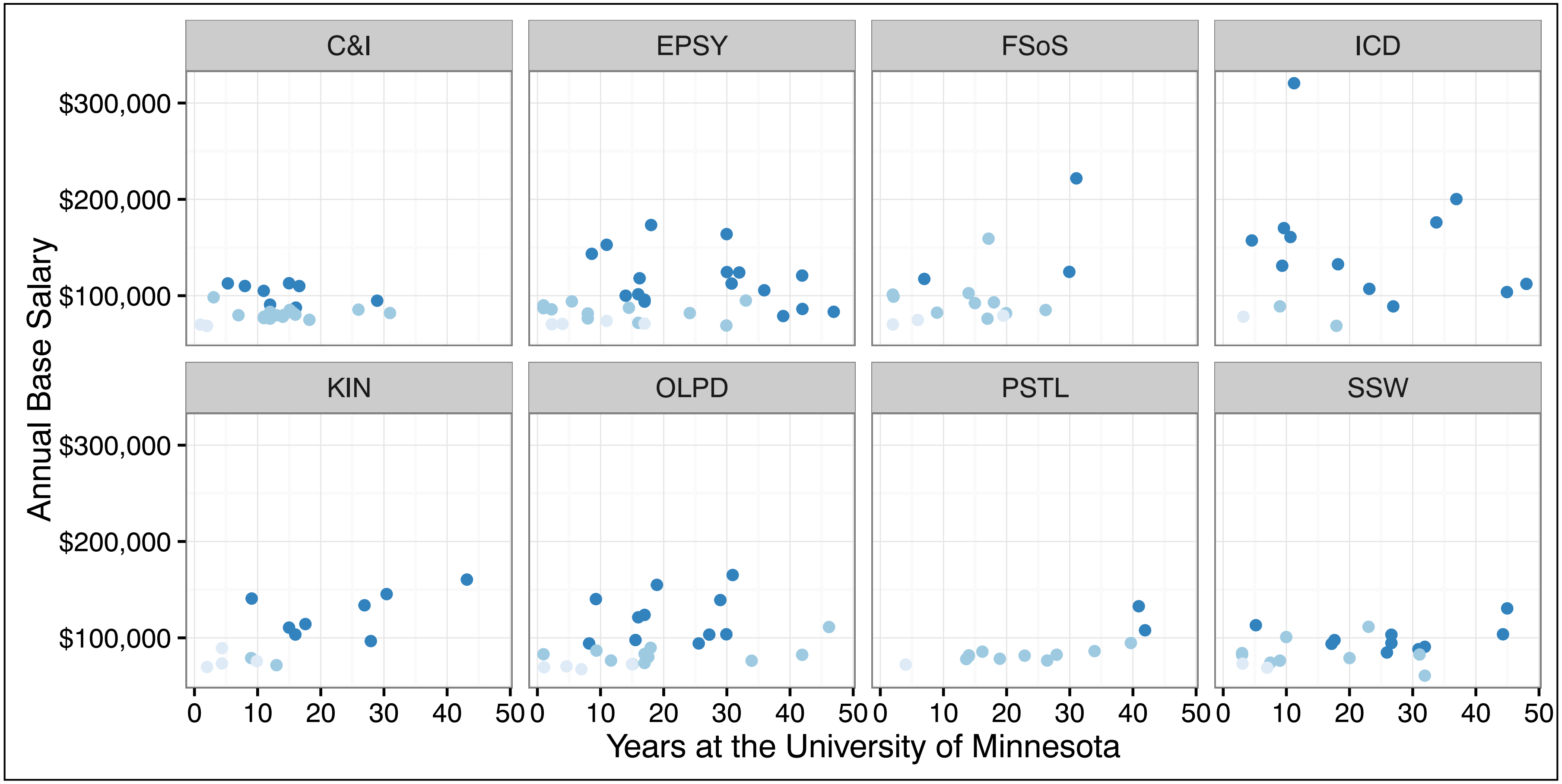


Figure 1. Relationship between annual base salary (in U.S. dollars) and years of experience at the University of Minnesota for 2015 CEHD faculty members. This relationship is shown for all eight faculty-hiring departments. The three sequential colors represent the three levels of faculty, namely, Associate Professor (lightest blue); Assistant Professor (medium blue), and Professor (darkest blue).

`#protip`: It is easier to use a word-processor (e.g., Word) to add the figure title and caption than to try and get it formatted correctly using R.

`#protip`: When you only have a few colors, include them in the caption rather than as a legend if you have space limits.

ggplot Resources

- **ggplot2 Cheatsheet***:* A one-page (front and back) cheatsheet of ggplot2 syntax with pictures https://www.rstudio.com/wp-content/uploads/2015/08/ggplot2-cheatsheet.pdf

- **ggplot2 Extensions***:* Third-party and user contributed extensions for some pretty cool plots http://www.ggplot2-exts.org/index.html

- **Cookbook for R:** Web-based version of Winston Chang's R Graphics Cookbook **http://www.cookbook-r.com/Graphs/** (The UMN library has electronic access to the actual book. Just search for "R Graphics Cookbook" and log-in with your x500.)

- **extrafonts package:** Use almost any font on your computer in your plots. http://blog.revolutionanalytics.com/2012/09/how-to-use-your-favorite-fonts-in-r-charts.html

`#protip`**:** Use Google to find out how to do just about anything with ggplot.