

# Computational Skills for Biostatistics I: Lecture 4

Amy Willis, Biostatistics, UW

October 19, 2017

# Graphical communication in practice

Graphical communication is critical for both *exploring* and *explaining* data

- ▶ base graphics in R: static
- ▶ ggplot: static
- ▶ shiny: interactive
- ▶ Non-data-based figures

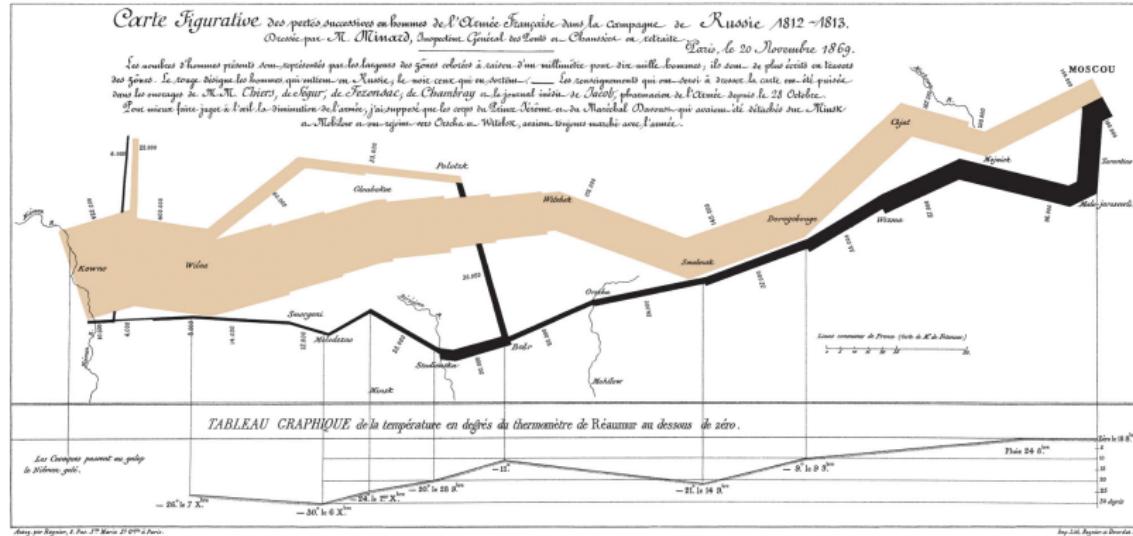
# Name to know: Edward Tufte

"If you're not doing something different, you're not doing anything at all."

The Rules:

1. Show Your Data
2. Use Graphics
3. Avoid Chartjunk
4. Utilize Data-ink
5. Use Labels
6. Utilize Micro/Macro
7. Separate Layers
8. Use Multiples
9. Utilize Color
10. Understand Narrative

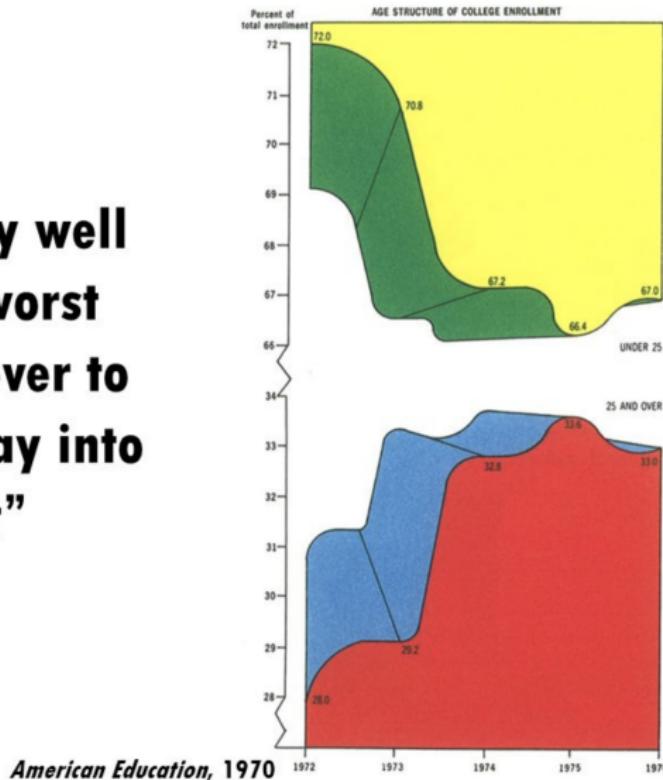
# The greatest statistical graphic ever drawn



*Fabry par Bayard, 8 Rue St-Martin 27 100 à Paris.*

# The worst statistical graphic ever drawn

**"This may well  
be the worst  
graphic ever to  
find its way into  
print"**



## ggplot

- ▶ ggplot is a function available in the `ggplot2` package.
- ▶ Graphics are built in layers: a plot is initialised, *then* data is then drawn, *then* annotations are added.
- ▶ Annotations include
  - ▶ scales
  - ▶ labels
  - ▶ legends
  - ▶ coordinate systems

# ggplot

ggplot {ggplot2}

R Documentation

## Create a new ggplot

### Description

`ggplot()` initializes a ggplot object. It can be used to declare the input data frame for a graphic and to specify the set of plot aesthetics intended to be common throughout all subsequent layers unless specifically overridden.

### Usage

```
ggplot(data = NULL, mapping = aes(), ..., environment = parent.frame())
```

### Arguments

<code>data</code>	Default dataset to use for plot. If not already a <code>data.frame</code> , will be converted to one by <a href="#"><code>fortify</code></a> . If not specified, must be supplied in each layer added to the plot.
<code>mapping</code>	Default list of aesthetic mappings to use for plot. If not specified, must be supplied in each layer added to the plot.
<code>...</code>	Other arguments passed on to methods. Not currently used.
<code>environment</code>	If a variable defined in the aesthetic mapping is not found in the data, <code>ggplot</code> will look for it in this environment. It defaults to using the environment in which <code>ggplot()</code> is called.

### Details

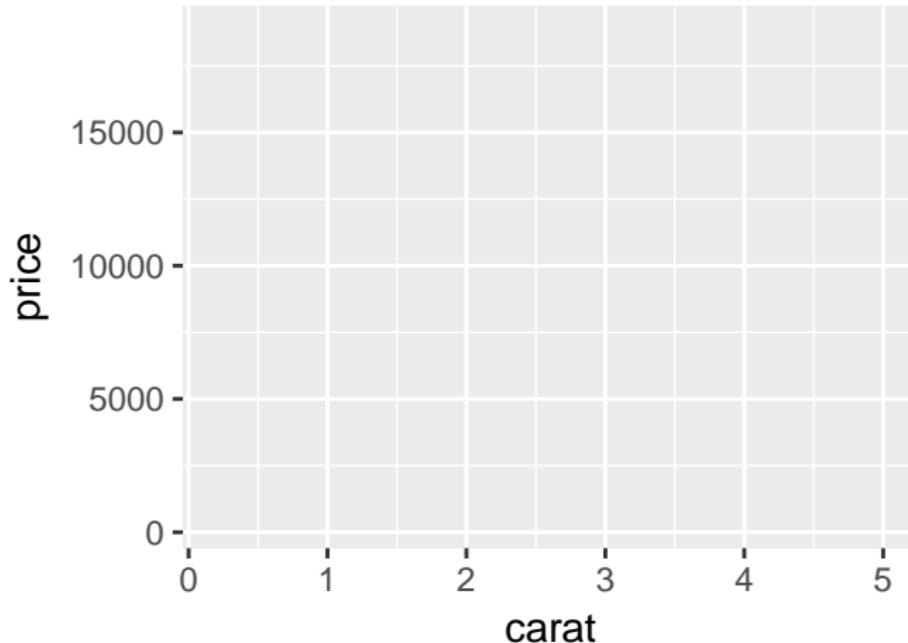
`ggplot()` is used to construct the initial plot object, and is almost always followed by `+` to add component to the plot. There are three common ways to invoke `ggplot`:

1. `ggplot(df, aes(x, y, <other aesthetics>))`
2. `ggplot(df)`
3. `ggplot()`

The first method is recommended if all layers use the same data and the same set of aesthetics, although this method can also be used to add a layer using data from another data frame. See the first example below. The second method specifies the default data frame to use for the plot, but no aesthetics are defined up front. This is useful when one data frame is used predominantly as layers are added, but the aesthetics may vary from one layer to another. The third method initializes a skeleton `ggplot` object which is fleshed out as layers are added. This method is useful when multiple data frames are used to produce different layers, as is often the case in complex graphics.

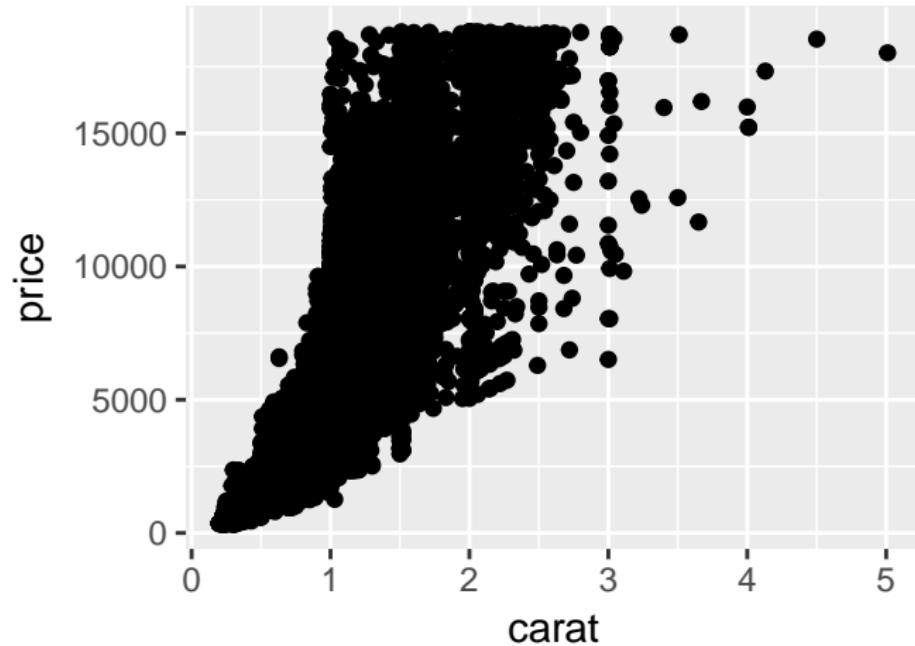
# ggplot

```
ggplot(diamonds, aes(x = carat, y = price))
```



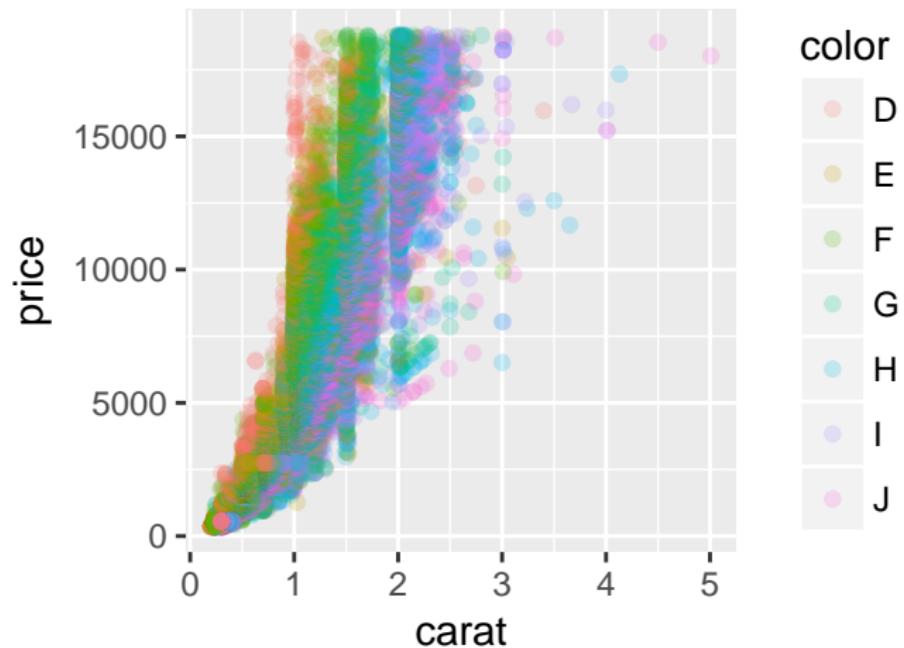
## Initialize, *then* add plotting elements

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point()
```



## Customise features about the layer

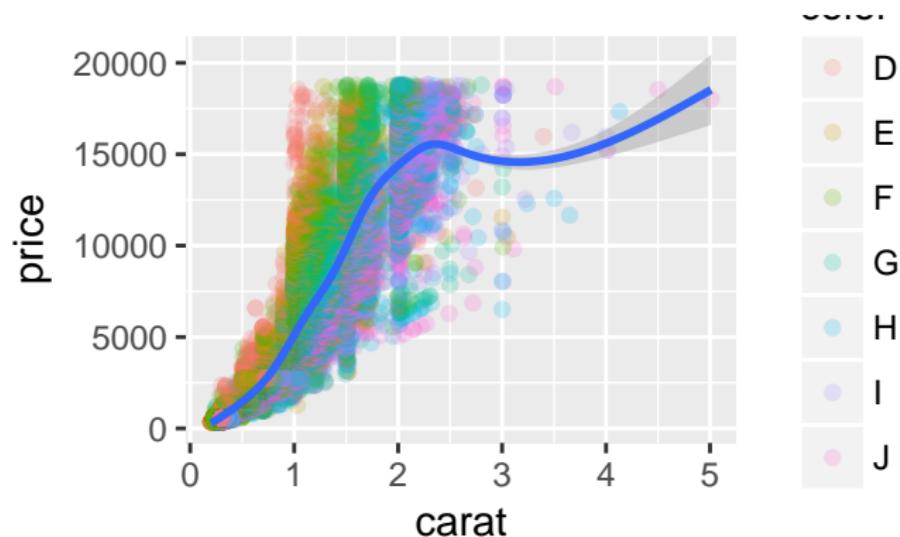
```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point(aes(col = color), alpha = 0.2)
```



## Add another layer

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point(aes(col = color), alpha = 0.2) + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam'
```



ggplot usually gives sensible results, but, e.g., the authors are not going to be experts in smoothing algorithms. Exercise caution and use your judgement!

## Equivalent calls

```
ggplot(diamonds, aes(x = carat, y = price)) + geom_point()  
ggplot(diamonds) + geom_point(aes(x = carat, y = price))  
ggplot(diamonds, aes(x = carat)) + geom_point(aes(y = price))
```

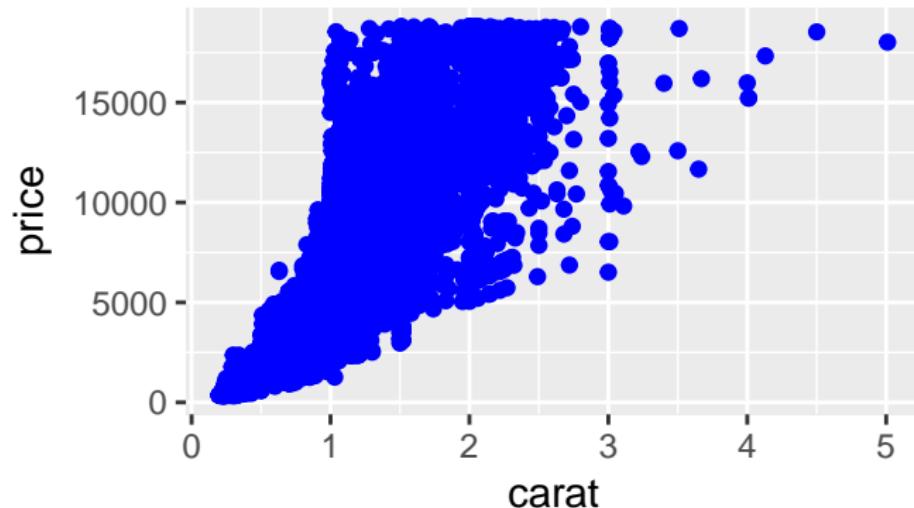
## Layering objects

Arguments to a ggplot call:

- ▶ The first argument to a ggplot is the data frame (or tibble)
- ▶ You can fix the aesthetics with aes() OR you can add aesthetics in layers
- ▶ Any aspect of a plot that you want to vary based on a variable needs to be wrapped in a aes() call

## aes() calls

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point(col = "blue")
```

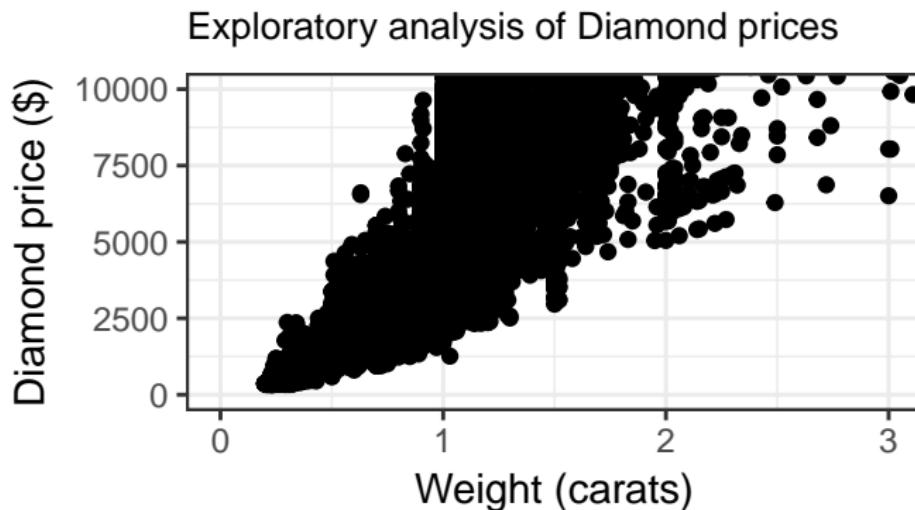


Color only exists locally (in the context of the diamonds), so the following results in an error:

```
# ggplot(diamonds, aes(x=carat, y=price))+geom_point(col=color)
```

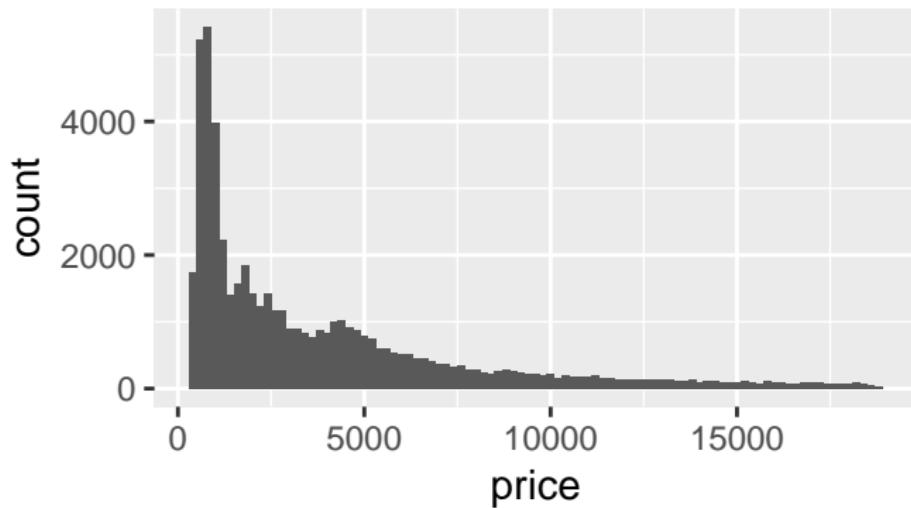
## More layers

```
ggplot(diamonds, aes(x = carat, y = price)) +  
  geom_point() + theme_bw() +  
  labs(x="Weight (carats)", y="Diamond price ($)") +  
  ggtitle("Exploratory analysis of Diamond prices") +  
  coord_cartesian(xlim = c(0, 3), ylim=c(0, 10000)) +  
  theme(plot.title = element_text(size = 10))
```



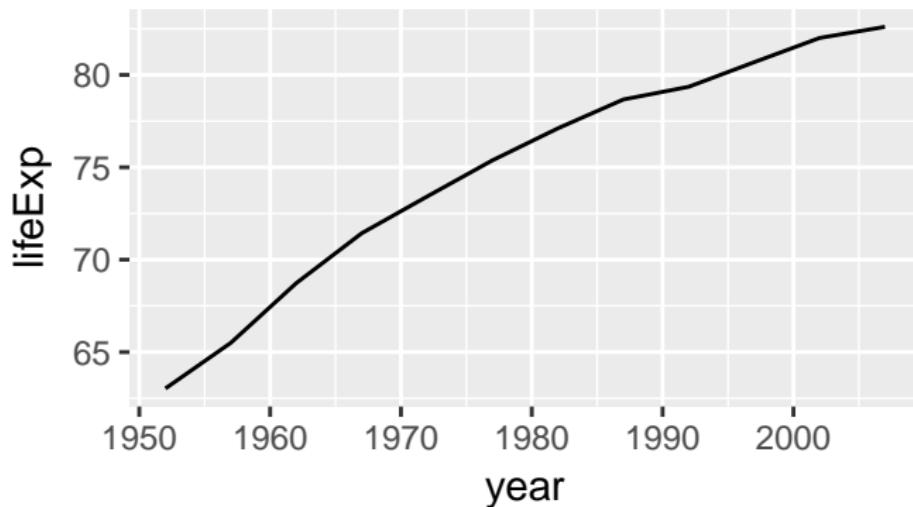
## Histograms

```
ggplot(diamonds, aes(x = price)) +  
  geom_histogram(binwidth = 200)
```



## Timeseries

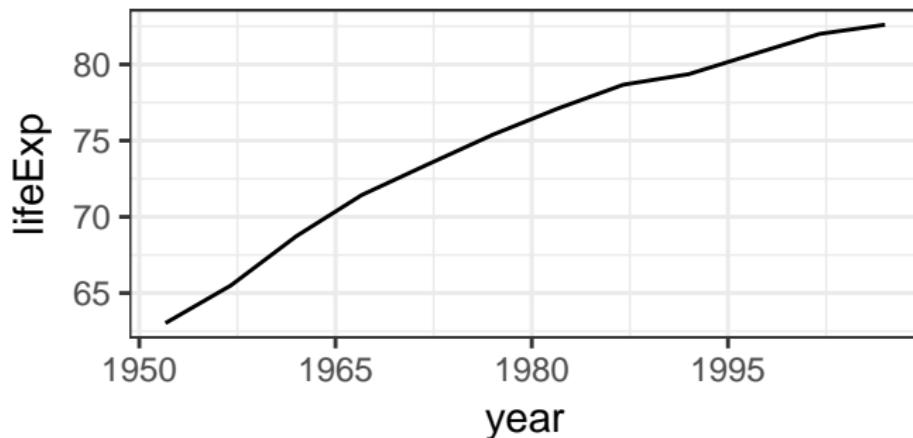
```
g <- ggplot(gapminder %>% filter(country == "Japan"),  
             aes(x = year, y = lifeExp)) + geom_line()  
g
```



## Storing and modifying

```
g + scale_x_continuous(breaks = seq(1950, 2011, 15)) +  
  theme_bw() +  
  ggtitle("Life expectancy in 20th C Japan")
```

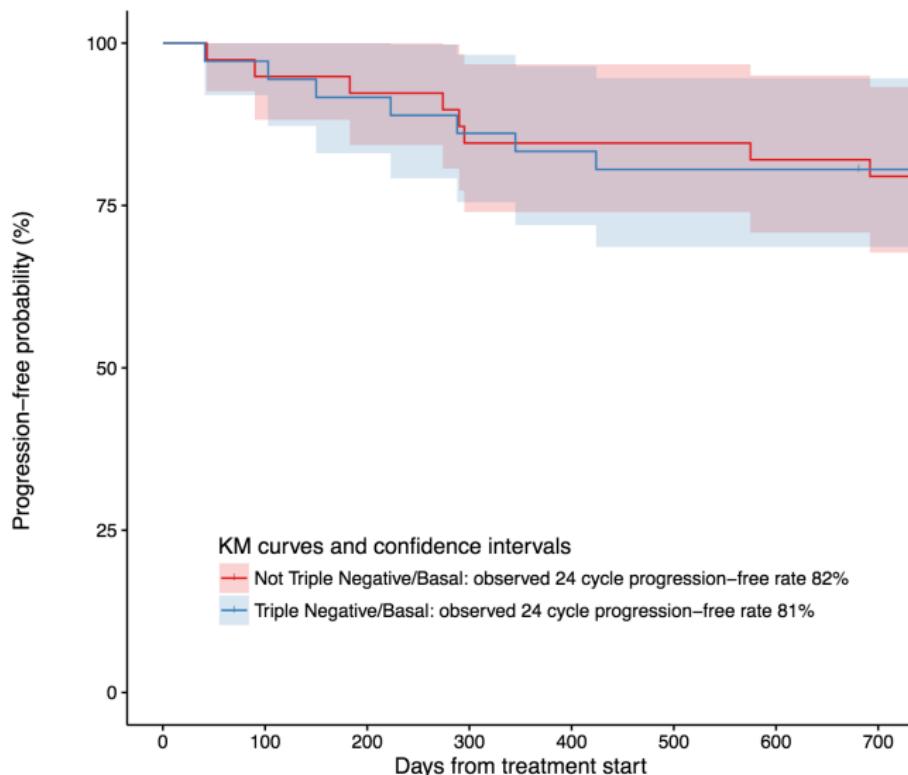
Life expectancy in 20th C Japan



Creating and plotting ggplot objects is not always fast; this may help

## Showing intervals using transparency

The most important information should be the most clear, the next most important information should be the next most clear, etc..



## Final tips

- ▶ Never, ever show a pie chart to a statistical audience... or the internet!
  - ▶ Exploding or otherwise
- ▶ Align plots with similar axes
- ▶ Defer less important points to appendices
  - ▶ Use figures where appropriate. Would a table suffice?
- ▶ Critique your own figures
  - ▶ Especially before public talks/conferences

Highly recommended reading:

“Graphical Display of Quantitative Information” by Edward Tufte

## Limitations of ggplot

- ▶ *Forces you to use good coding practice*
  - ▶ Data should all come from the same data frame
  - ▶ Painful at first but better than a retraction!
- ▶ Constrained to two-dimensional (see `plot3d`, `scatterplot3d`)
- ▶ No graphs/trees (`igraph`)
- ▶ Static (`shiny`)
- ▶ Limited use for non-statistical graphics (Illustrator, Photoshop, Inkscape)

## Coming up

- ▶ Reminder: `git pull` – any time you work on any file in a directory with version control!
- ▶ Homework 4 due next Thursday at the usual time in the usual way
- ▶ `shiny` with Katie Wilson