

Week 12: Data Visualization

EMSE 6574 | John Paul Helveston | November 11, 2019

Getting started

- 1) Download the `week12notes.zip` file for class today (link in `slack/classroom`).
- 2) Make sure you have the "tidyverse" installed and loaded:

```
library(tidyverse)
```

- 3) Load two data frames:

```
birds  
bears
```

The Challenger disaster

On January 28, 1986 the space shuttle Challenger exploded



The Challenger disaster

NASA Engineers had the data on temperature & o-ring failure

TEMPERATURE CONCERN ON
SRM JOINTS

27 JAN 1986

HISTORY OF O-RING DAMAGE ON SRM FIELD JOINTS

SRM No.	Cross Sectional View			Top View		
	Erosion Depth (in.)	Perimeter Affected (deg)	Nominal Dia. (in.)	Length Of Max Erosion (in.)	Total Heat Affected Length (in.)	Clocking location (deg)
22A	None	None	0.280	None	None	36°--66°
22A	None	None	0.280	None	None	338°-18°
15A	0.010	154.0	0.280	4.25	5.25	163
15B	0.038	130.0	0.280	12.50	58.75	314
15B	None	45.0	0.280	None	29.50	324
41D RH Forward Field	0.028	110.0	0.280	3.00	None	275
41C LH Aft Field*	None	None	0.280	None	None	—
41B LH Forward Field	0.040	217.0	0.280	3.00	14.50	351
STS-2 RH Aft Field	0.053	116.0	0.280	--	--	90

*Hot gas path detected in putty. Indication of heat on O-ring, but no damage.
**Soot behind primary O-ring.
***Soot behind primary O-ring, heat affected secondary O-ring.

Clocking location of leak check port - 0 deg.

OTHER SRM-15 FIELD JOINTS HAD NO BLOHOLENS IN PUTTY AND NO SOOT NEAR OR BEYOND THE PRIMARY O-RING.
SRM-22 FORWARD FIELD JOINT HAD PUTTY PATH TO PRIMARY O-RING, BUT NO O-RING EROSION AND NO SOOT BLOWBY. OTHER SRM-22 FIELD JOINTS HAD NO BLOHOLENS IN PUTTY.

BLOW BY HISTORY

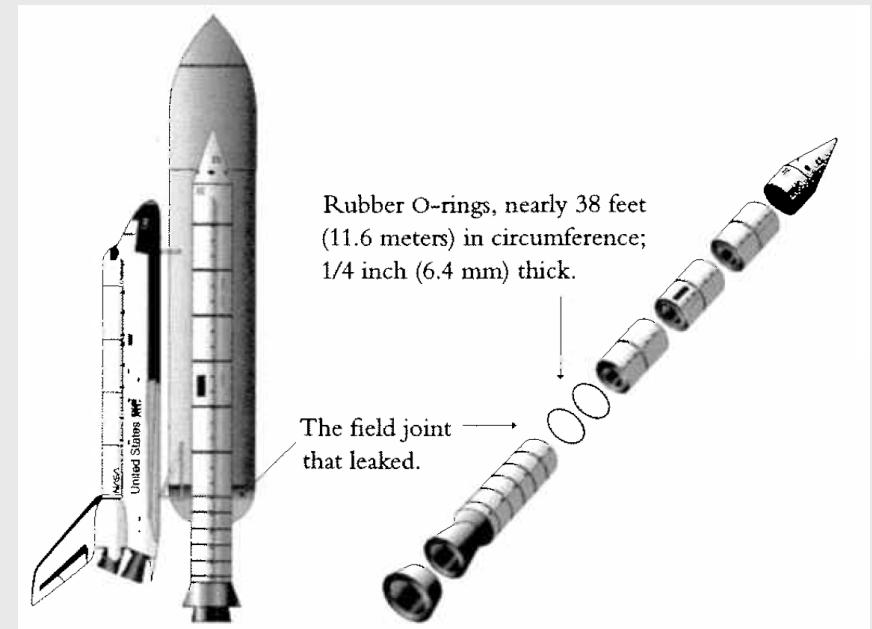
SRM-15 WORST BLOW-BY
o 2 CASE JOINTS (30°), (110°) ARC
o MUCH WORSE VISUALLY THAN SRM-22

SRM-22 BLOW-BY
o 2 CASE JOINTS (30-40°)

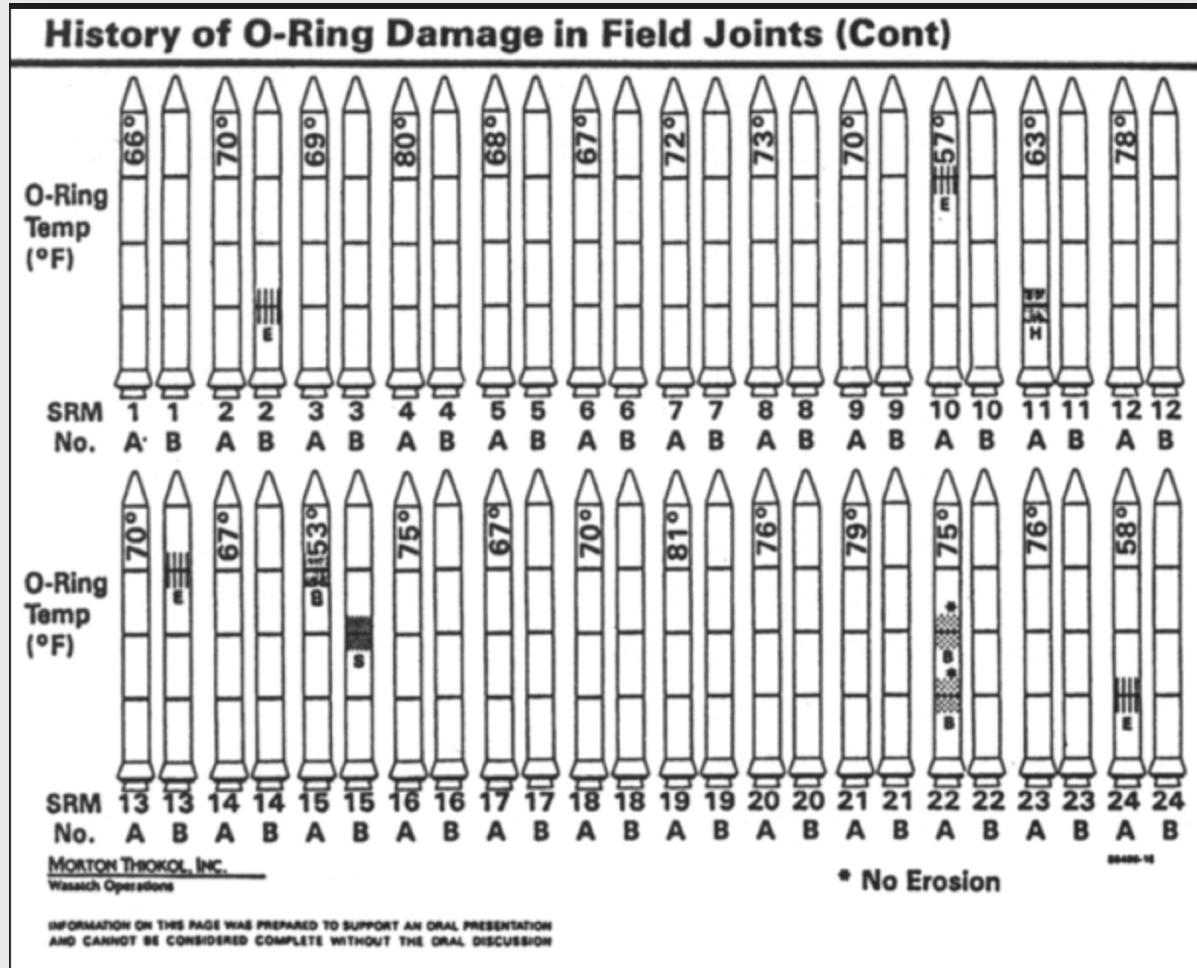
SRM-17, 15, 16A, 18, 23A, 24A
o NOZZLE Blow-by

HISTORY OF O-RING TEMPERATURES (DEGREES - F)

MOTOR	MST	AMB	O-RING	WIND	MOTOR	O-RING
DM-4	68	36	47	10 MPH	DM-4	47
DM-2	76	45	52	10 MPH	DM-2	52
QM-3	72.5	40	48	10 MPH	QM-3	48
QM-4	76	48	51	10 MPH	QM-4	51
SRM-15	52	64	53	10 MPH	SRM-15	53
SRM-22	77	78	75	10 MPH	SRM-22	75
SRM-25	55	26	29	10 MPH	SRM-25	29
			27	25 MPH		

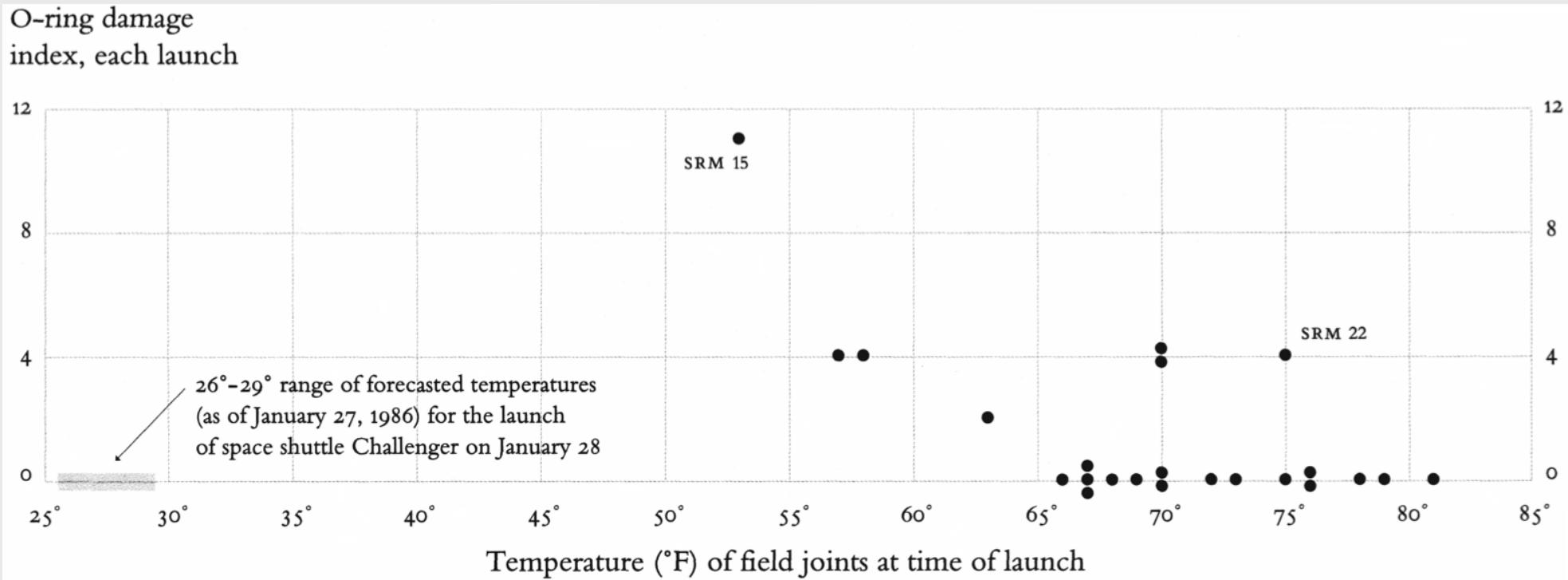


What NASA was shown



Tufte, Edward R. (1997) *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, Cheshire, Connecticut.

What NASA *should* have been shown



Tufte, Edward R. (1997) *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, Cheshire, Connecticut.

Today's data:

Bear attacks in North America

Explore the `bears` data frame:

```
glimpse(bears)  
head(bears)
```

Two basic plots in R

(there are others)

Scatterplots



Histograms



Scatterplots with `plot()`

Plot relationship between two variables

General syntax:

```
plot(x = <x_vector>, y = <y_vector>)
```

Scatterplots with `plot()`

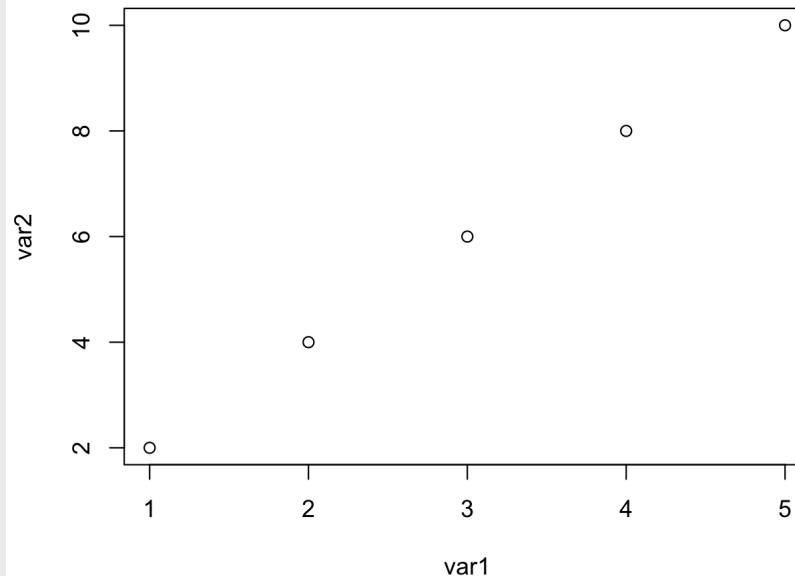
Plot relationship between two variables

General syntax:

```
plot(x = <x_vector>, y = <y_vector>)
```

Example:

```
var1 <- 1:5  
var2 <- 2*var1  
plot(x = var1, y = var2)
```



Scatterplots with `plot()`

`x` and `y` must have the same length!

```
var2 <- var2[-1]
```

```
length(var1) == length(var2)
```

```
## [1] FALSE
```

```
plot(x = var1, y = var2)
```

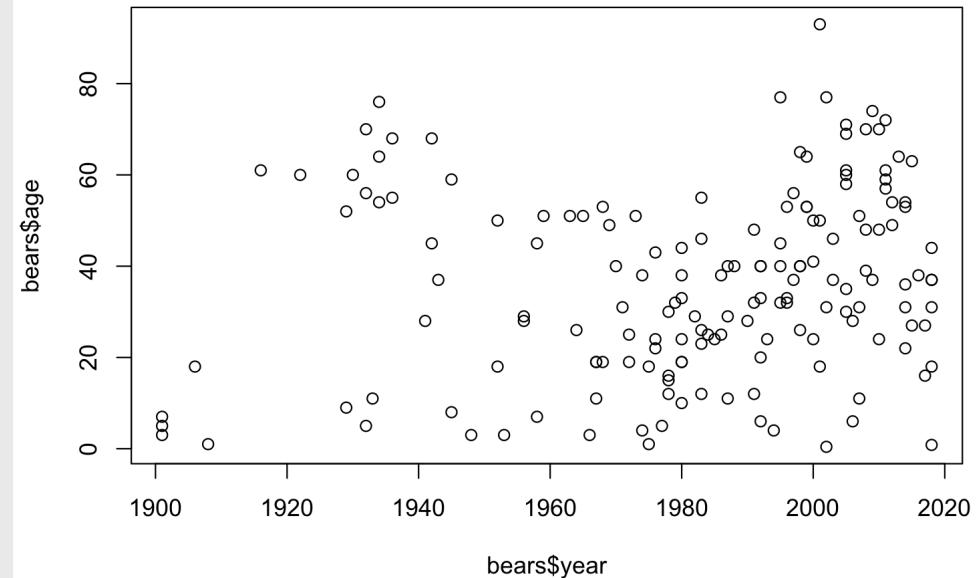
```
## Error in xy.coords(x, y, xlabel, ylabel, log): 'x' and 'y' lengths differ
```

Scatterplots with `plot()`

Plotting variables from a data frame:

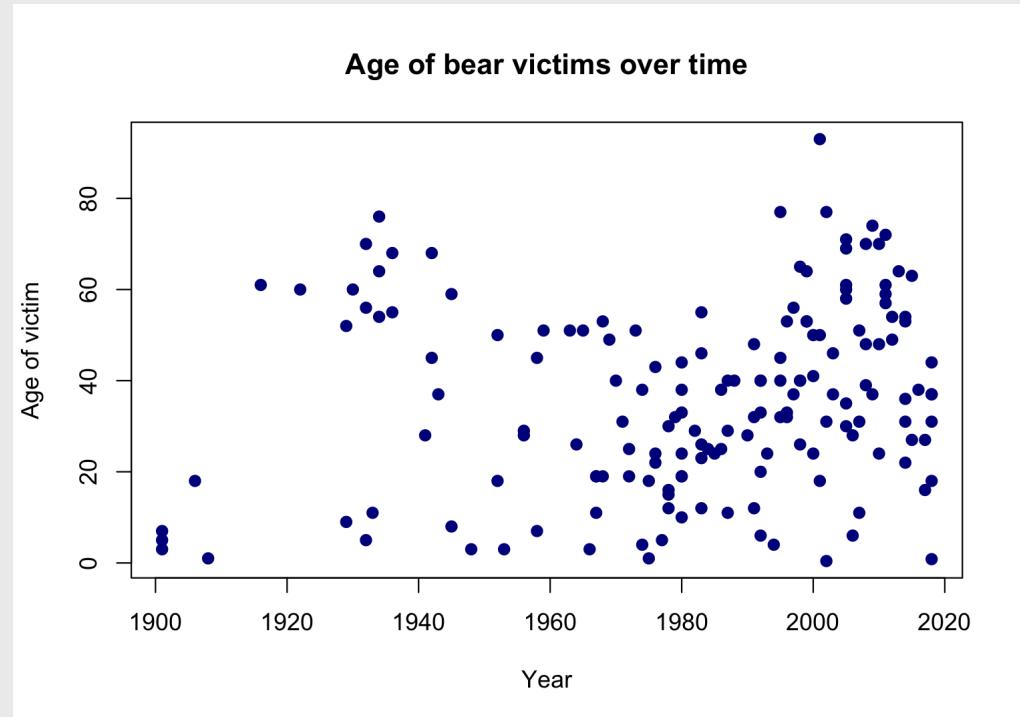
Plot `year` vs. `age`:

```
plot(x = bears$year, y = bears$age)
```



Making `plot()` pretty

```
plot(x      = bears$year,  
     y      = bears$age,  
     col    = 'darkblue', # Point color  
     pch   = 19, # Point shape  
     main  = "Age of bear victims over time",  
     xlab  = "Year",  
     ylab  = "Age of victim")
```

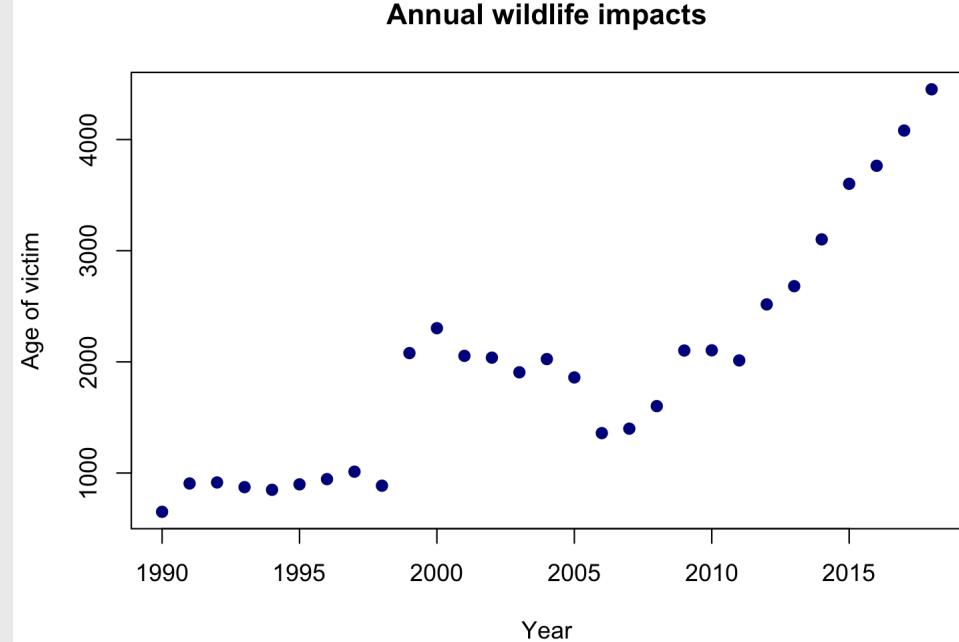


Practice: `plot()`

Does the annual number of bird impacts appear to be changing over time? (use the `birds` data frame)

```
annualCount <- birds %>%
  count(incident_year)

plot(
  x = annualCount$incident_year,
  y = annualCount$n,
  col = "darkblue",
  pch = 19,
  main = "Annual wildlife impacts",
  xlab = "Year",
  ylab = "Age of victim")
```



Histograms with `hist()`

Plot the *distribution* of a single variable

General syntax:

```
hist(x = <x_vector>)
```

Histograms with `hist()`

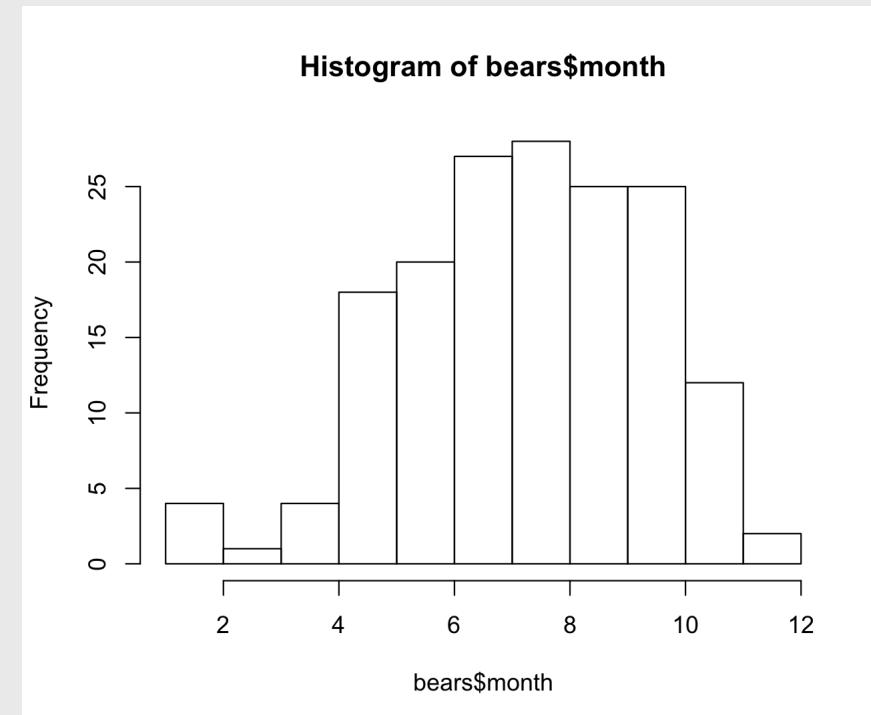
Plot the *distribution* of a single variable

General syntax:

```
hist(x = <x_vector>)
```

Example:

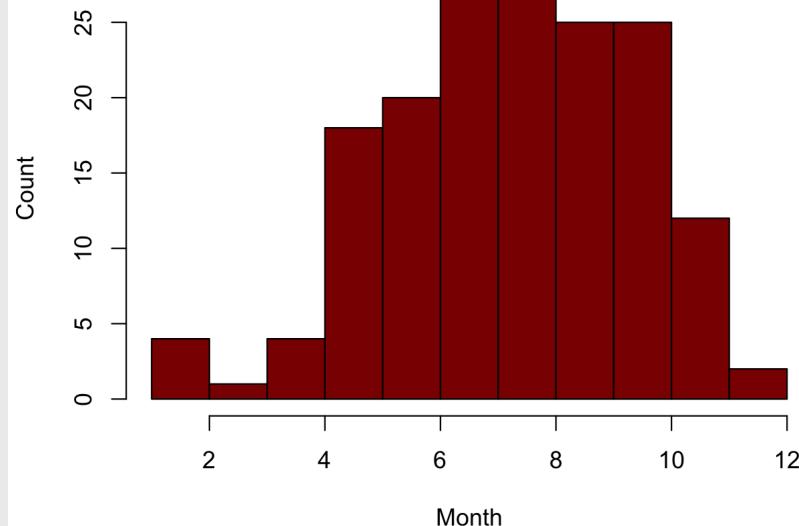
```
hist(bears$month)
```



Making `hist()` pretty

```
hist(x      = bears$month,  
     breaks = 12,  
     col    = 'darkred',  
     main   = "Distribution of bear killings by month",  
     xlab   = "Month",  
     ylab   = "Count")
```

Distribution of bear killings by month



Practice: `hist()`

Answer the following questions by visually examining the `birds` data frame:

- Which months have the highest and lowest number of bird impacts in the dataset?
- Which aircrafts experience more impacts: 2-engine, 3-engine, or 4-engine?
- At what height do most impacts occur?

Advanced figures with `ggplot2`



"Grammar of Graphics" (gg)

Concept developed by Leland Wilkinson, 1999

ggplot2 package developed by Hadley Wickham, 2005

Layer 1: The data

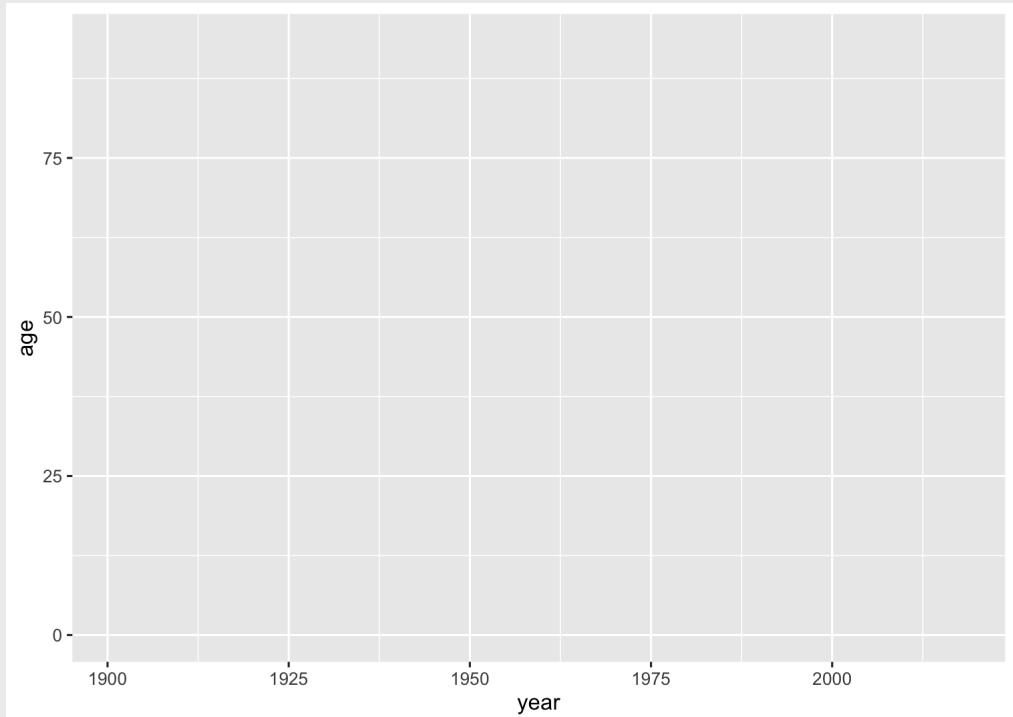
The `ggplot()` function initializes the plot:

```
ggplot(data = bears)
```

Layer 2: The aesthetics

The `aes()` function determines which variables will be *mapped* to the axes:

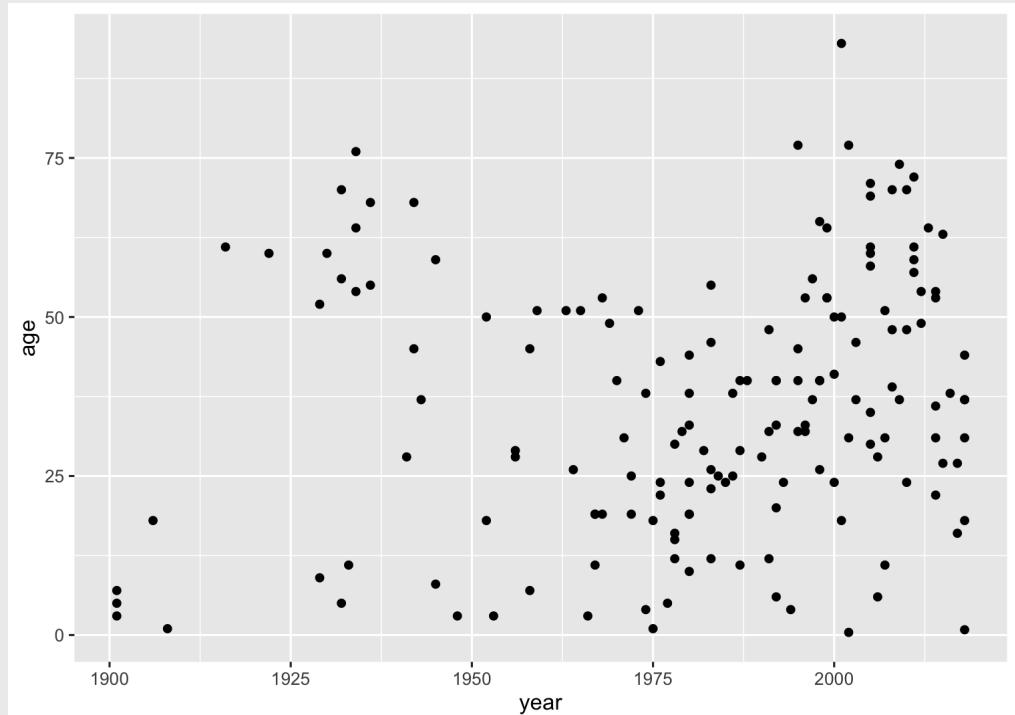
```
ggplot(data = bears, aes(x = year, y = age))
```



Layer 3: The geometries

Use  to add geometries (e.g. points) to the plot:

```
ggplot(data = bears, aes(x = year, y = age)) +  
  geom_point()
```



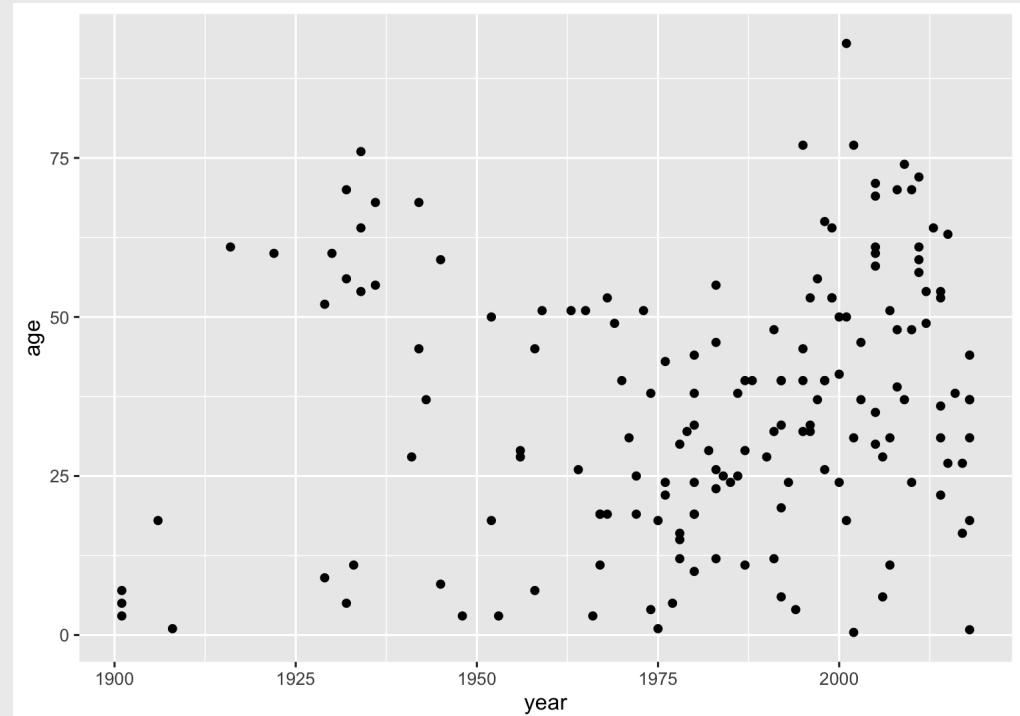
Other common geometries

- `geom_point()`: makes a scatter plot
- `geom_line()`: lines connecting data points
- `geom_bar()`: adds bars for bar charts and histograms
- `geom_boxplot()`: adds boxes for boxplots

Scatterplots with `geom_point()`

Example:

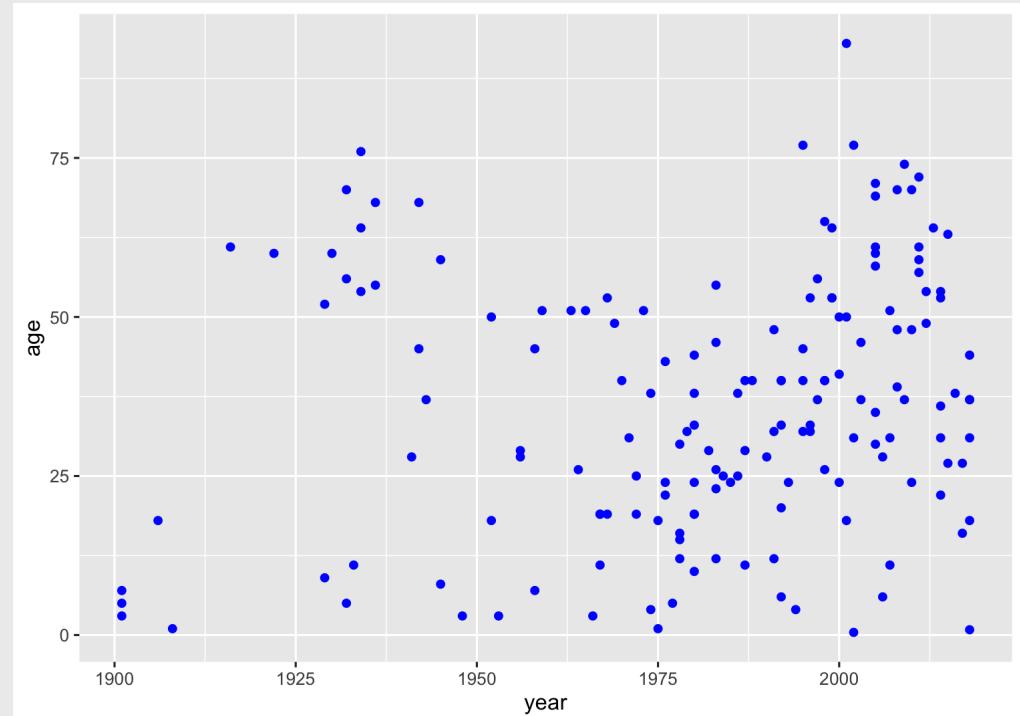
```
ggplot(data = bears, aes(x = year, y = age)) +  
  geom_point()
```



Scatterplots with `geom_point()`

Change the point color:

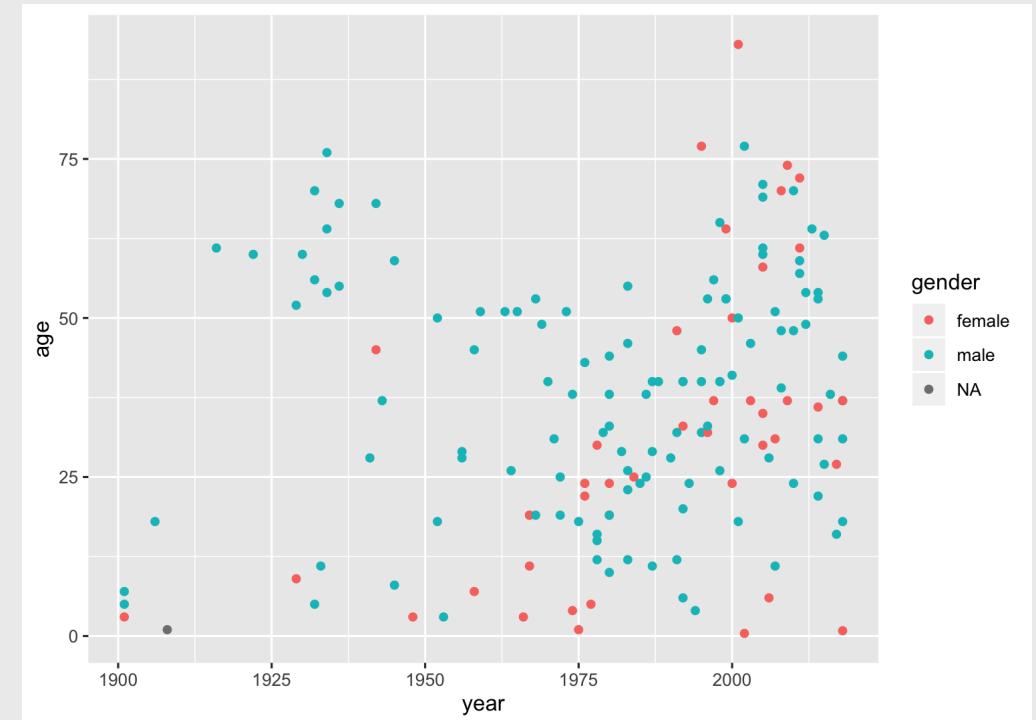
```
ggplot(data = bears, aes(x = year, y = age)) +  
  geom_point(color = 'blue')
```



Scatterplots with `geom_point()`

Change the point color based on another variable:

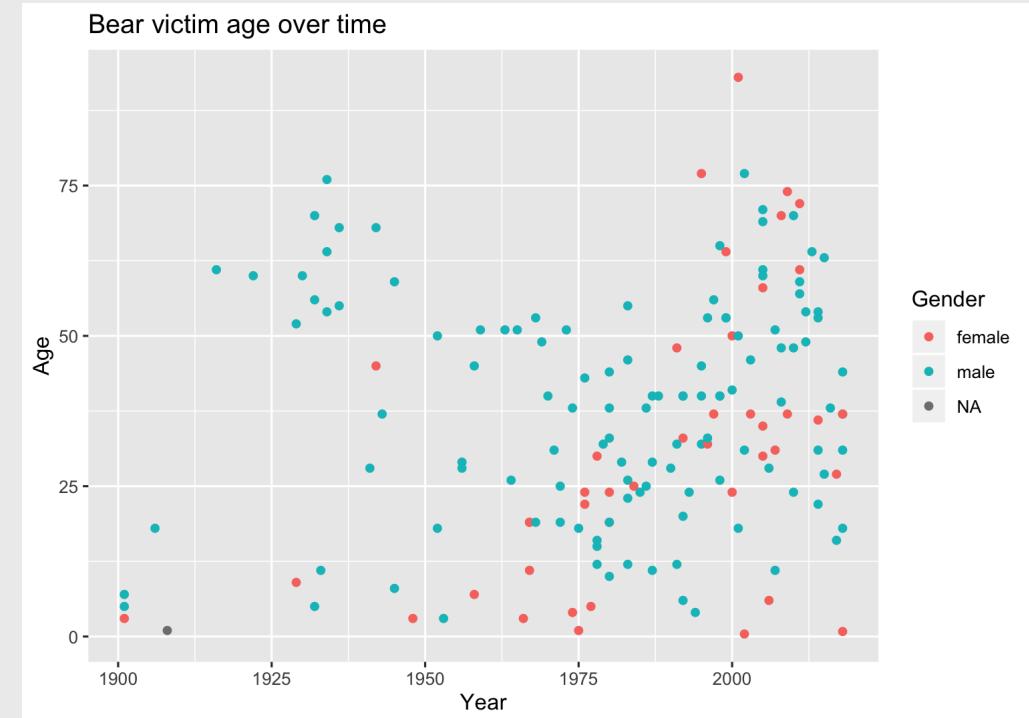
```
ggplot(data = bears, aes(x = year, y = age)) +  
  geom_point(aes(color = gender))
```



Scatterplots with `geom_point()`

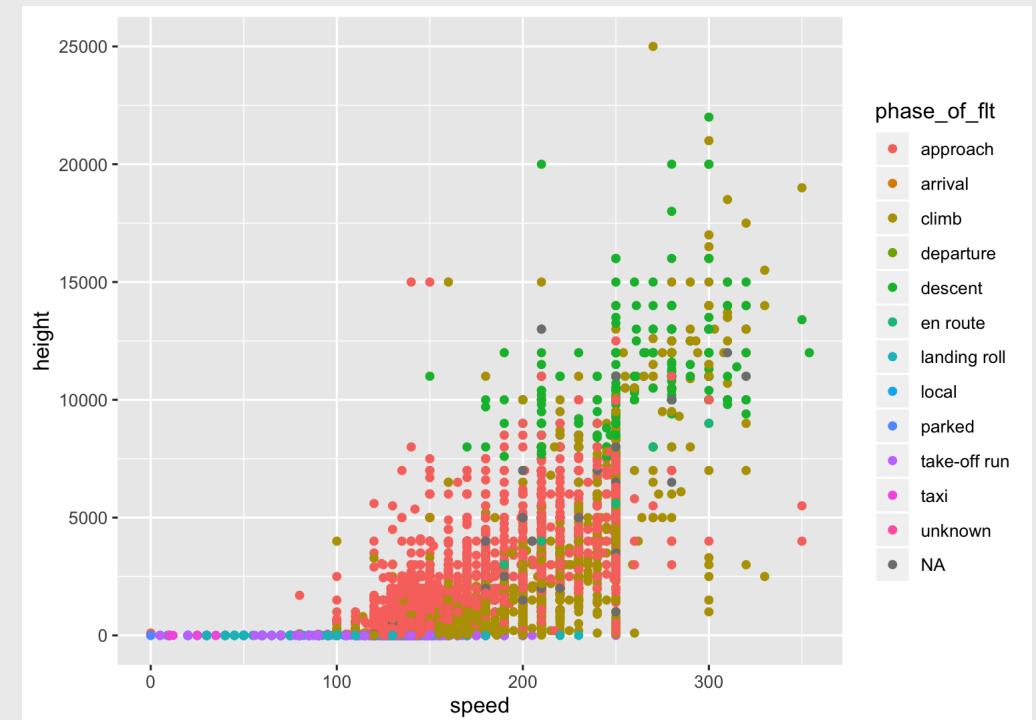
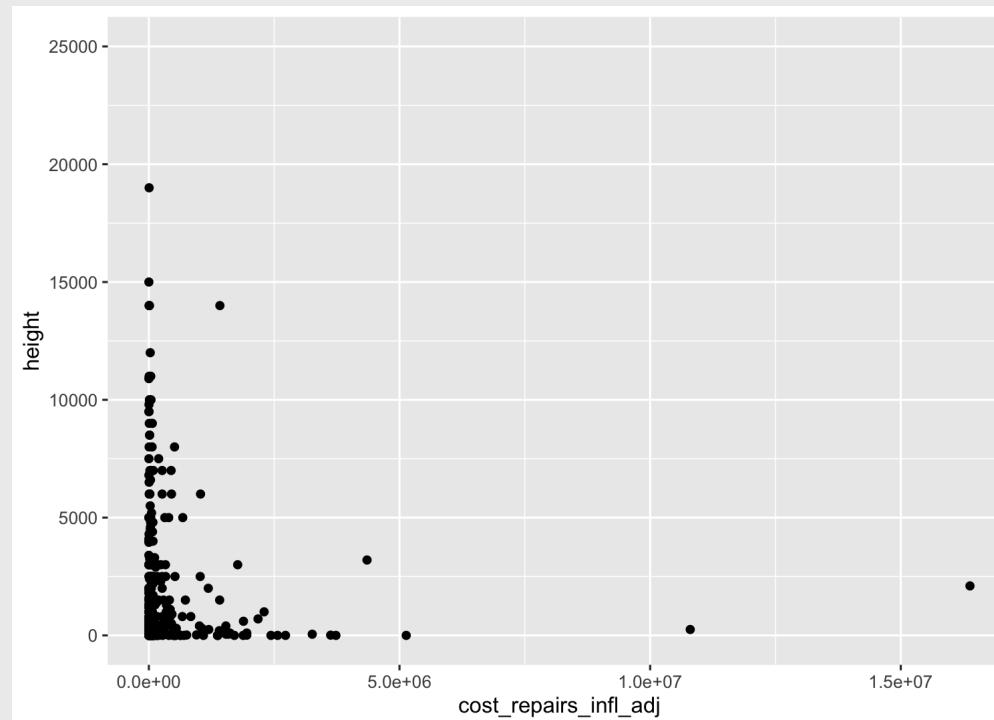
Adjust labels with `labs()` layer:

```
ggplot(data = bears, aes(x = year, y = age)) +  
  geom_point(aes(color = gender)) +  
  labs(x = "Year",  
       y = "Age",  
       title = "Bear victim age over time",  
       color = "Gender")
```



Practice: `geom_point()`

Use the `birds` data frame and `geom_point()` to create the following plots



5 minute break!

Stand up

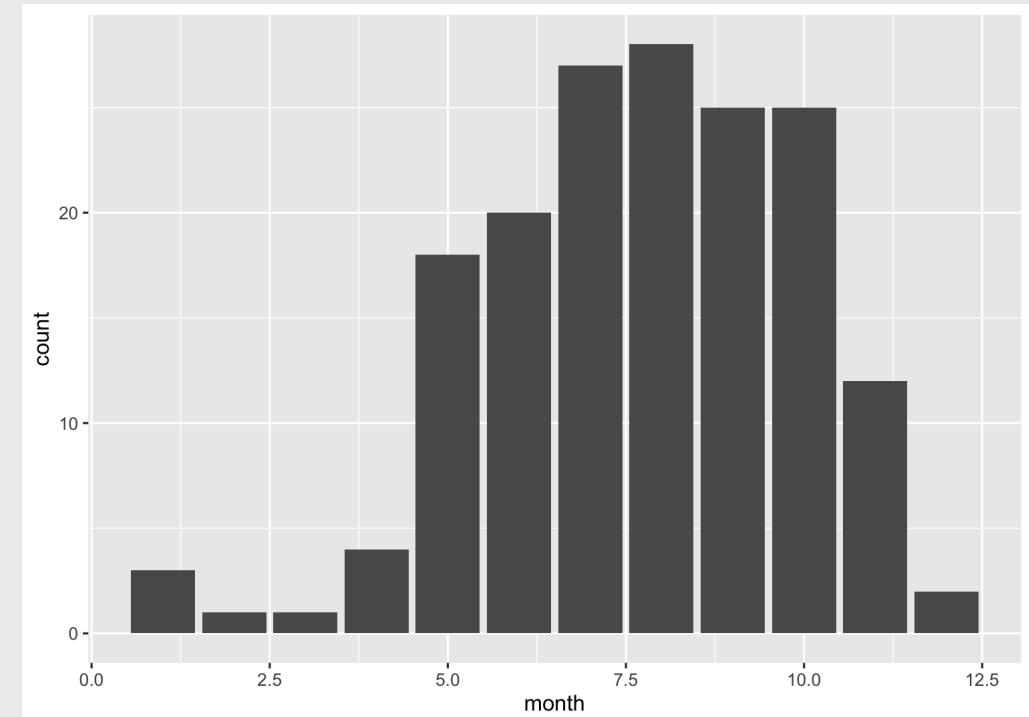
Move around

Stretch!

Histograms with `geom_bar()`

Example:

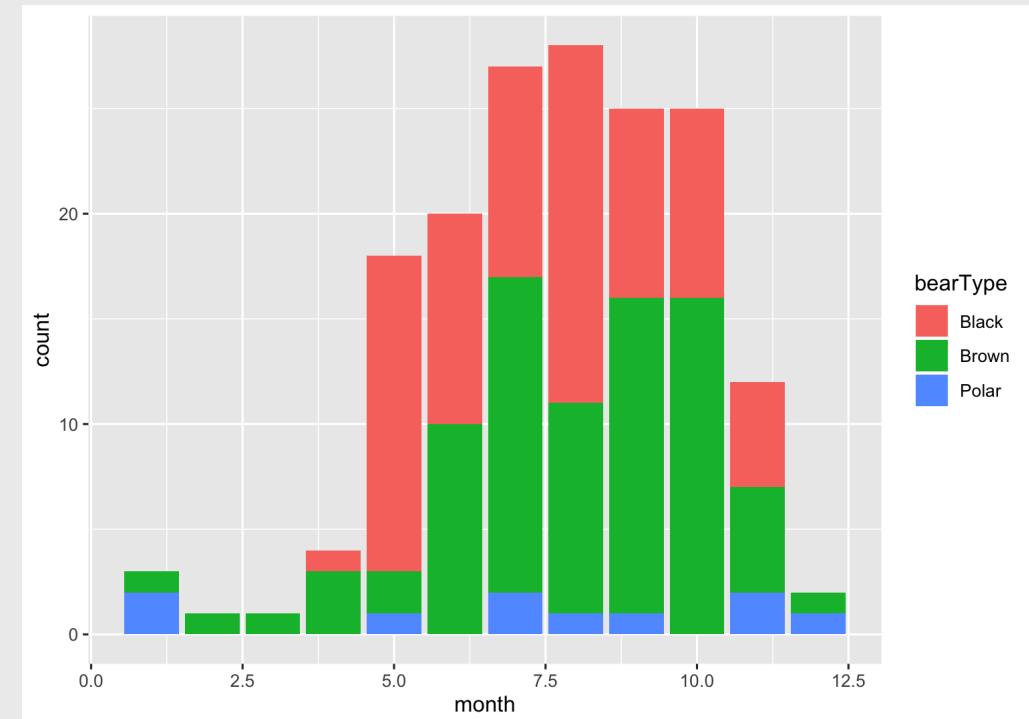
```
ggplot(data = bears, aes(x = month)) +  
  geom_bar()
```



Histograms with `geom_bar()`

Use `fill` (not `color`) to change the color of the bars:

```
ggplot(data = bears, aes(x = month)) +  
  geom_bar(aes(fill = bearType))
```



Bar chars with `geom_bar()`

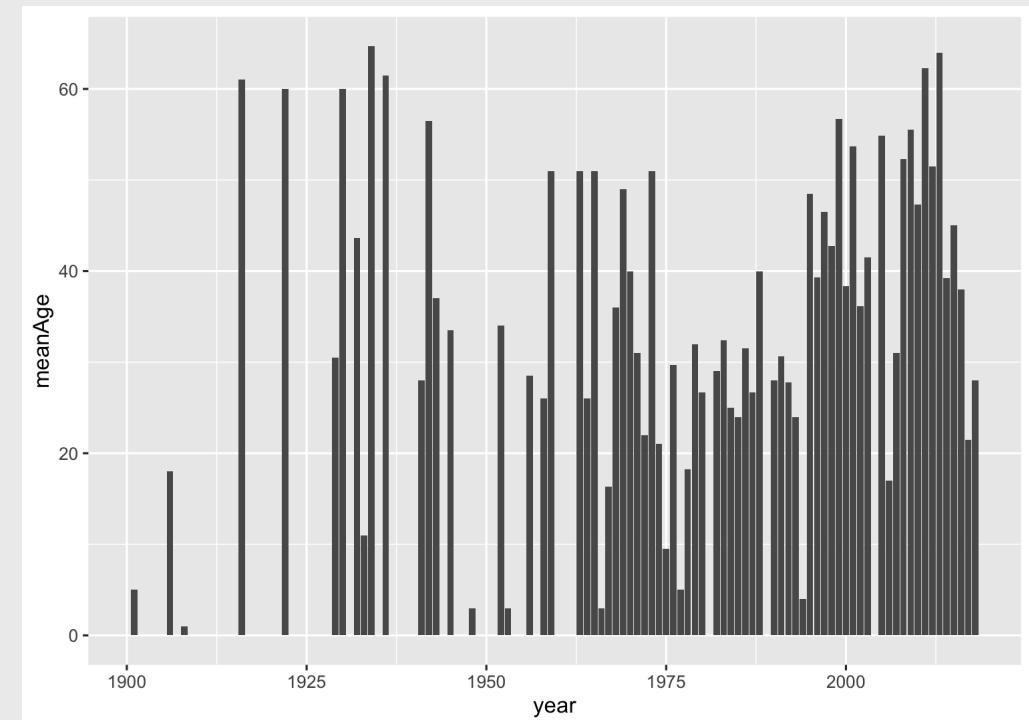
Use both `x` and `y` variables:

```
meanAgeAnnual <- bears %>%
  filter(!is.na(age)) %>%
  group_by(year) %>%
  summarise(meanAge = mean(age))

ggplot(data=meanAgeAnnual, aes(x=year, y=meanAge))
  geom_bar(stat = 'identity')
```

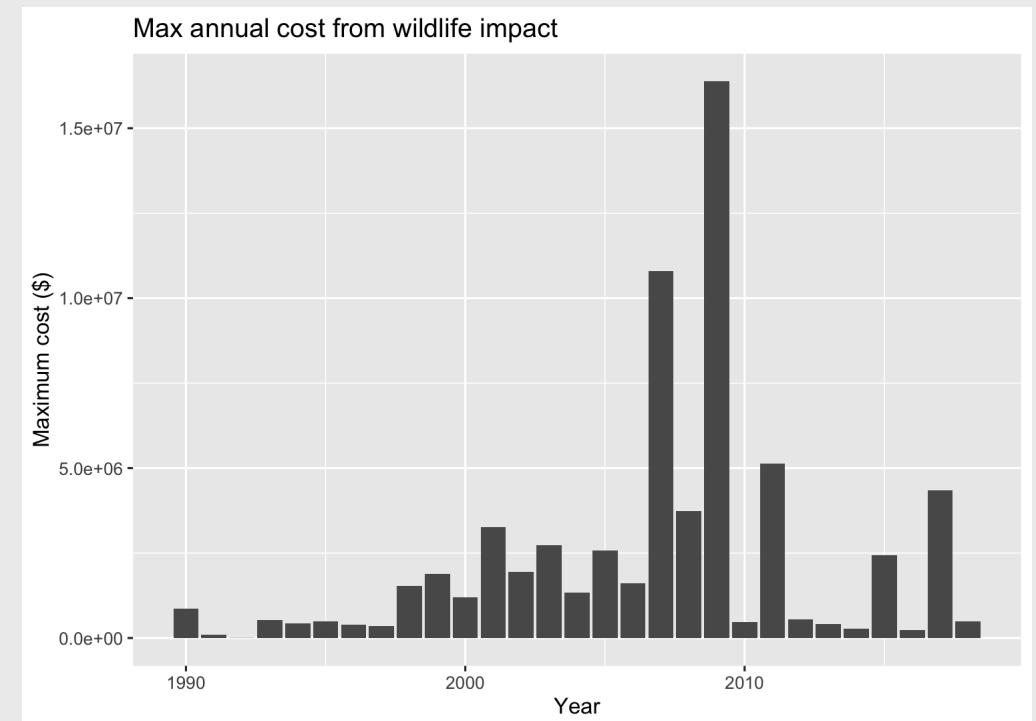
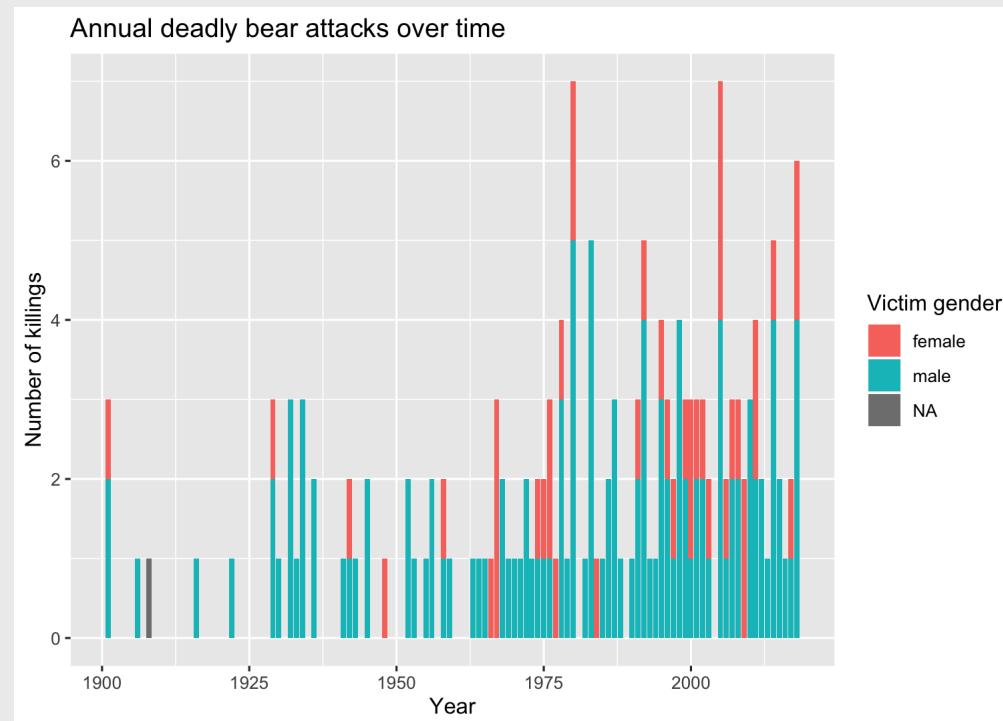
Note addition of `stat = 'identity'`

(Default "stat" is `count`)



Practice: `geom_bar()`

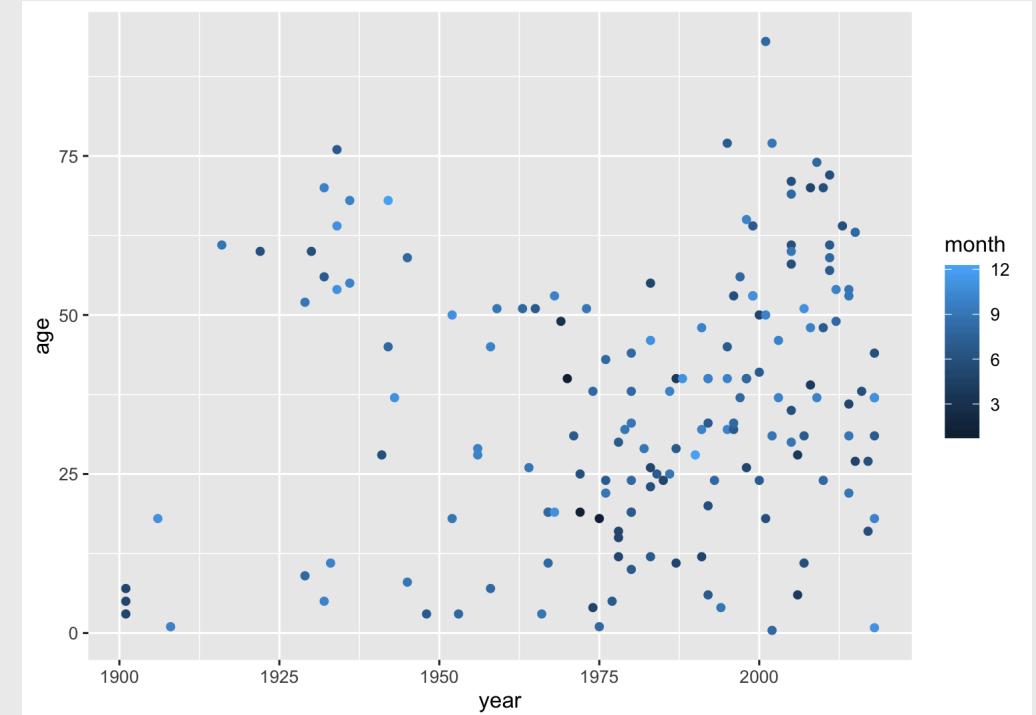
Use the `birds` data frame and `geom_bar()` to create the following plots



"Factors" = Categorical variables

By default, R makes numeric variables *continuous*

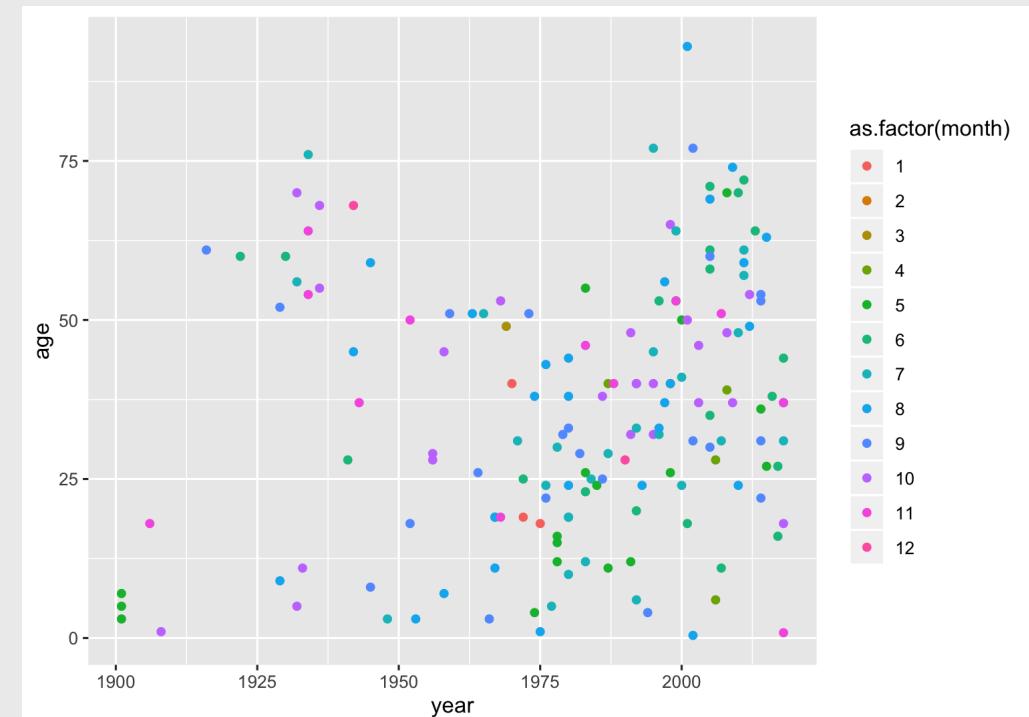
```
ggplot(bears, aes(x=year, y=age)) +  
  geom_point(aes(color=month))
```



"Factors" = Categorical variables

You can make a continuous variable *categorical* using `as.factor()`

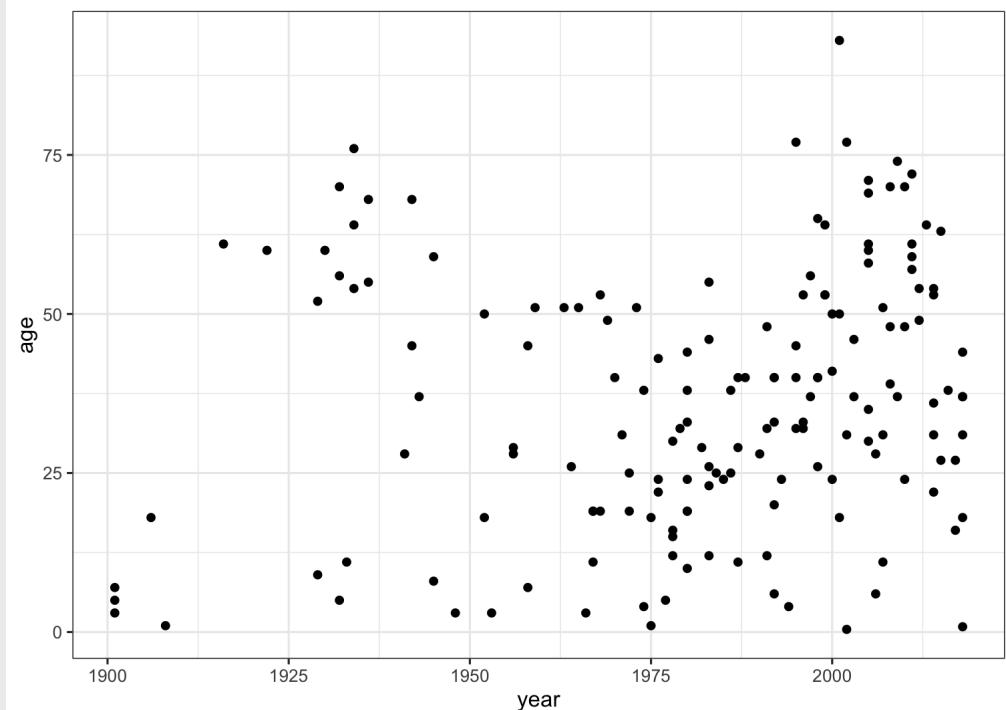
```
ggplot(bears, aes(x=year, y=age)) +  
  geom_point(aes(color=as.factor(month)))
```



Working with themes

Themes change *global*/features of your plot, like the background color, grid lines, etc.

```
ggplot(data = bears, aes(x = year, y = age)) +  
  geom_point() +  
  theme_bw()
```



Some common themes

Very commonly used:

- `theme_bw`: My personal default
- `theme_minimal`: Another simple, minimal theme
- `theme_classic`: Classic R plotting style
- `theme_void`: Completely blank

Others:

- `theme_light`
- `theme_dark`
- `theme_grey`
- `theme_gray`
- `theme_linedraw`
- `theme_test`

Getting some gg-assistance

```
install.packages("ggThemeAssist")
```

Save figures with `ggsave()`

First, assign the plot to an object name:

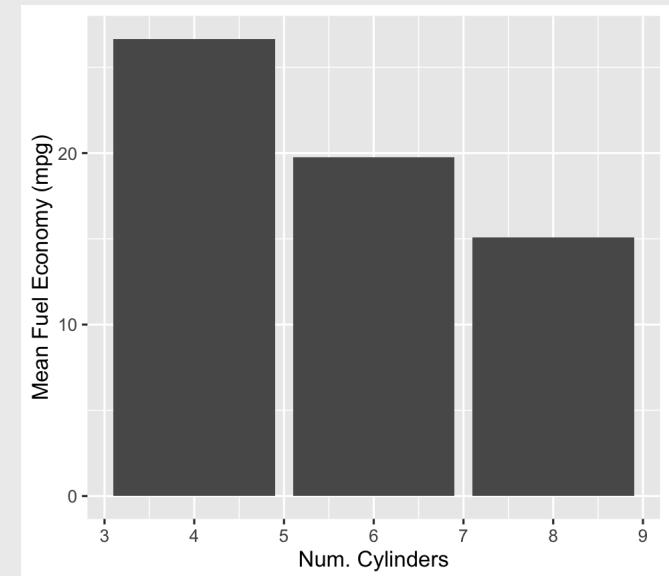
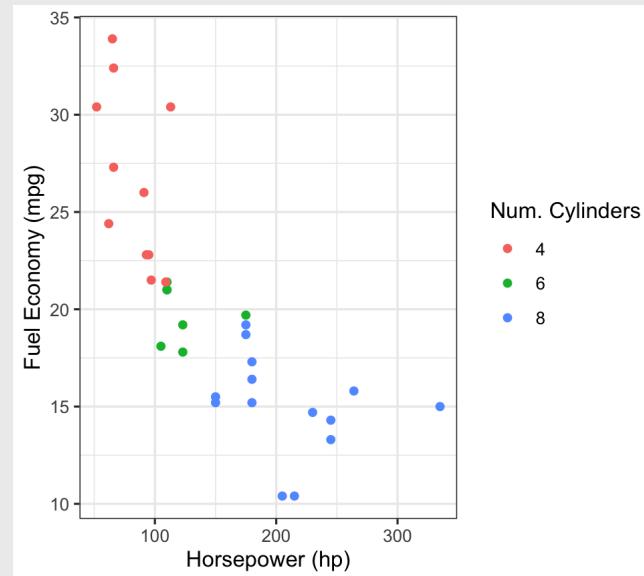
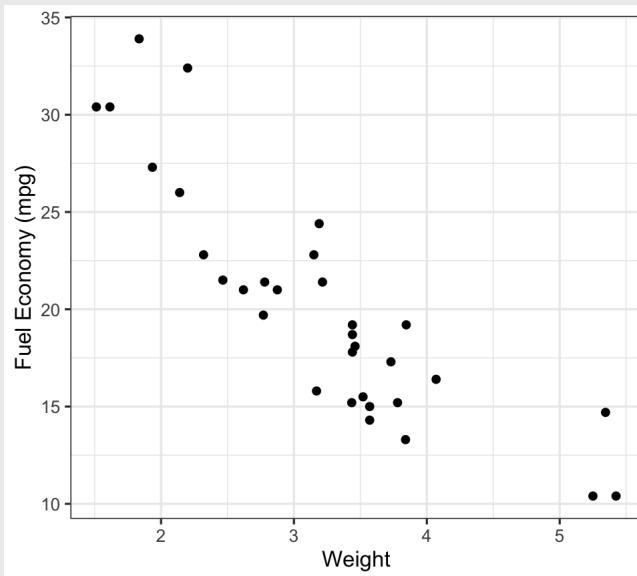
```
scatterPlot <- ggplot(data = df) +  
  geom_point(aes(x = year, y = age))
```

Then use `ggsave()` to save the plot:

```
ggsave(filename = file.path('data', 'scatterPlot.pdf'),  
       plot    = scatterPlot,  
       width   = 6,  
       height  = 4)
```

Extra practice 1

Use the `mtcars` data frame to create the following plots



Extra practice 2

Use the [mpg](#) data frame to create the following plot

