DS 116 -Data Visualization

Visualizing categorical data

Habet Madoyan

American University of Armenia

Univariate categorical variable

- The first and most common way to visualize categorical variables is barchart
- A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. These values usually are either absolute or relative frequencies of each variable.
- The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.

Univariate categorical variable

Lets look at the summer.csv, the dataset that contains information on summer Olympic games
Get a subset of data for USA only

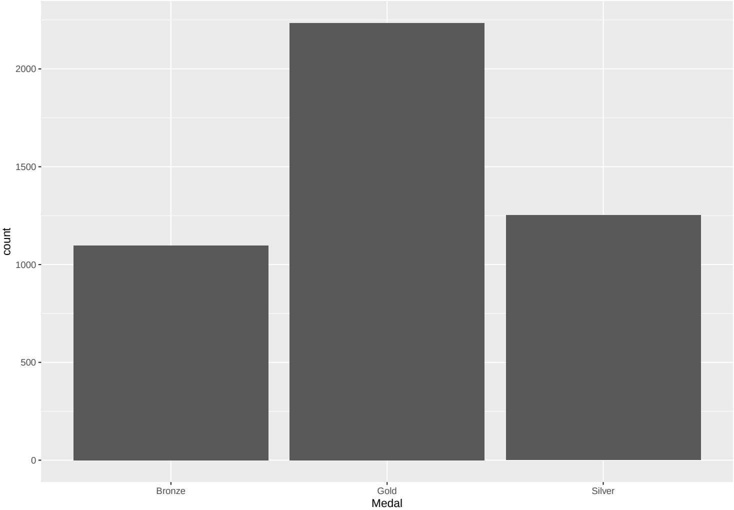
```
summer <- read.csv('Data/summer.csv')
summer_usa <- summer %>% filter(Country == 'USA')
```

Univariate categorical variable

- In ggplot2 bar chart is created with the geom_bar() layer.
- Note that you need to provide only one aesthetics - x,
- The y aesthetics is calculated via statistical transformation **count** (the default value)

Univariate categorical variable

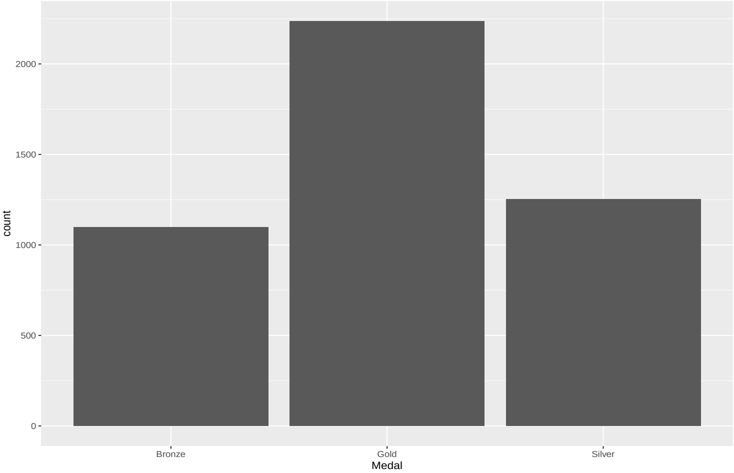
```
ggplot(summer_usa, aes(x = Medal)) + geom_bar(stat = 'count')
```



Univariate categorical variable

You could also use stat_count()

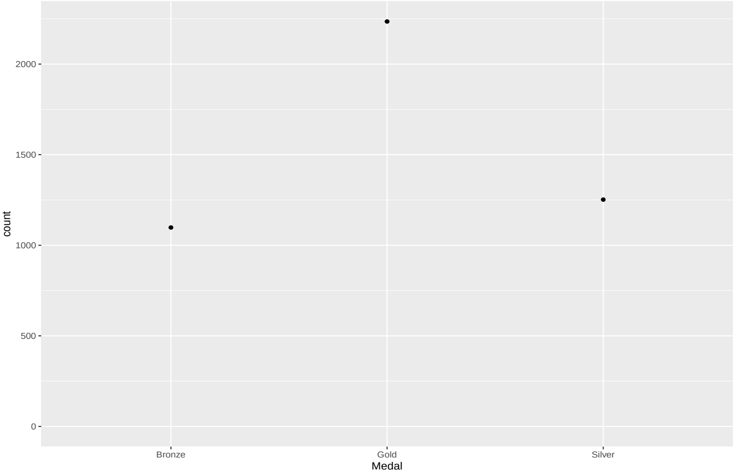
```
ggplot(summer_usa, aes(x = Medal)) + stat_count()
```



Univariate categorical variable

The difference is that with stat_count you can use other geom as well

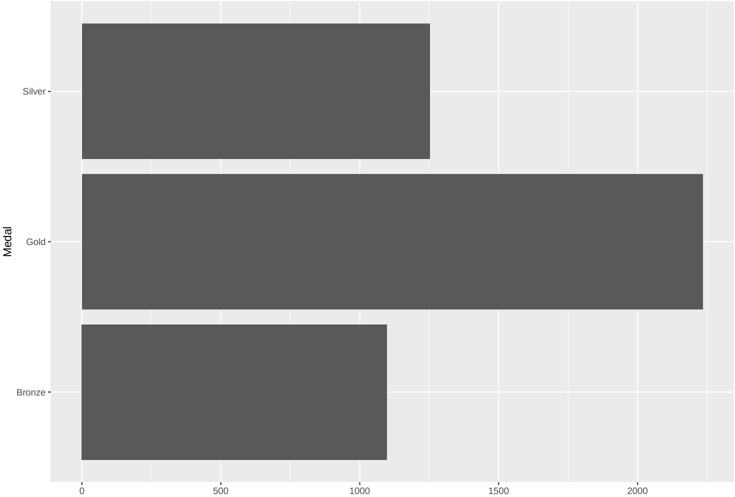
```
ggplot(summer_usa, aes(x = Medal)) + stat_count(geom = 'point')
```



Univariate categorical variable

You can get the vertical bar chart just setting y aesthetics instead of x

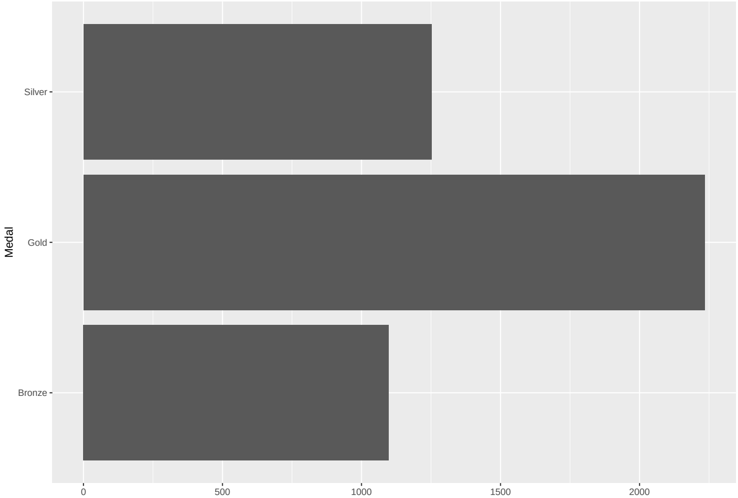
```
ggplot(summer_usa, aes(y = Medal)) + geom_bar()
```



Univariate categorical variable

Or use coord_flip()

```
ggplot(summer_usa, aes(x = Medal)) + geom_bar() + coord_flip()
```



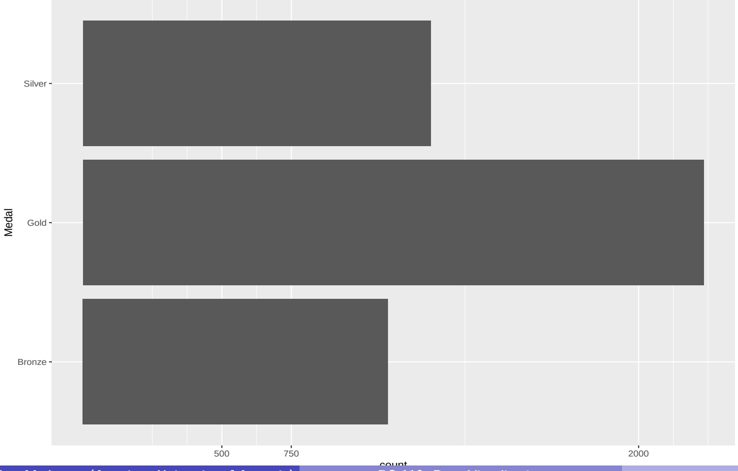
Univariate categorical variable

Note these two methods are not strictly equivalent

- The first one sets x aesthetics to count
- coord_flip() flips the coordinates

Univariate categorical variable

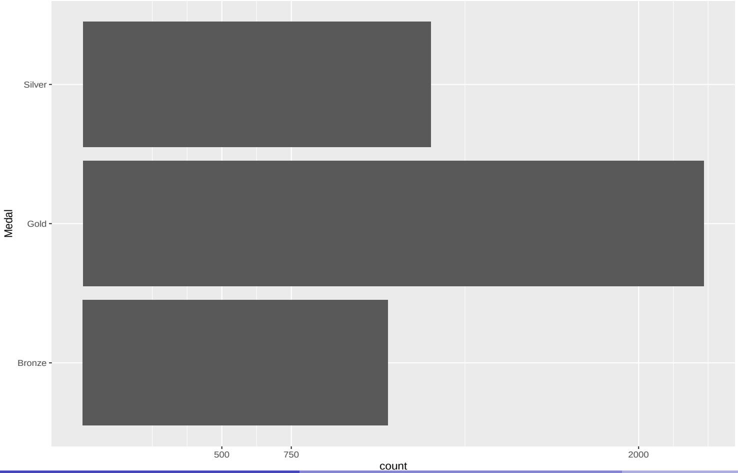
```
ggplot(summer_usa, aes(x = Medal)) + geom_bar() +
  coord_flip() +
  scale_y_continuous(breaks = c(500,750,2000))
```



Univariate categorical variable

But

```
ggplot(summer_usa, aes(y = Medal)) + geom_bar() +
  scale_x_continuous(breaks = c(500,750,2000))
```



Univariate categorical variable

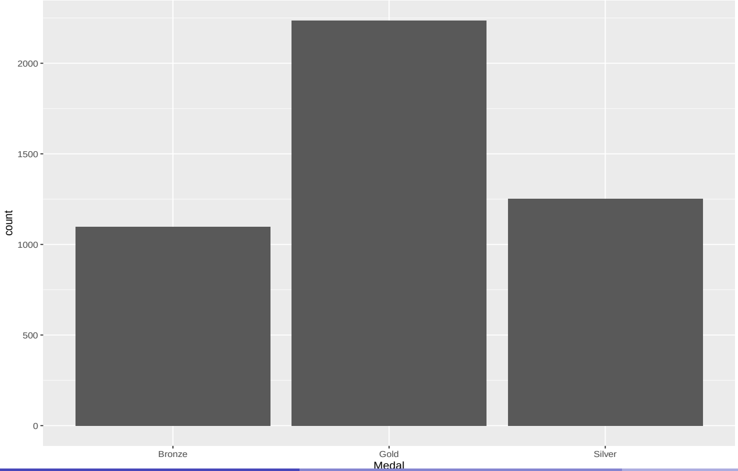
You can create a bar chart from already aggregated data, but need to change the statistical transformation

```
summer_usa %>% group_by(Medal) %>% summarise(count = n())
## # A tibble: 3 x 2
##   Medal  count
##   <chr>  <int>
## 1 Bronze  1098
## 2 Gold    2235
## 3 Silver  1252
```

Univariate categorical variable

And add y aesthetics

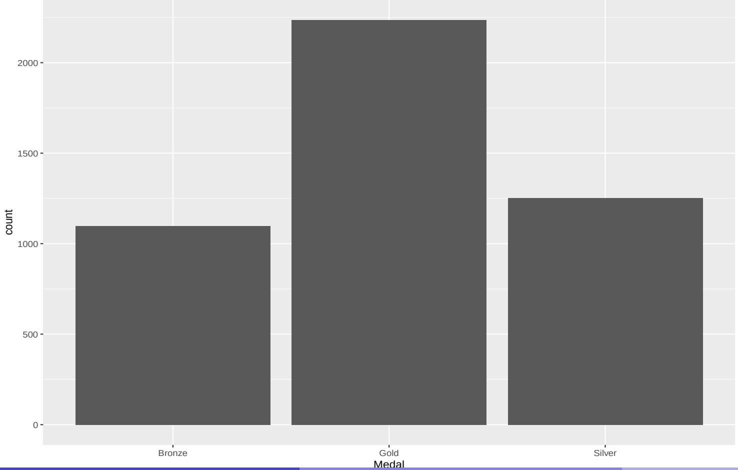
```
summer_usa %>% group_by(Medal) %>% summarise(count = n()) %>%
ggplot(aes(x = Medal, y = count)) + geom_bar(stat = 'identity')
```



Univariate categorical variable

Alternatively, when no statistical transformation is done, you can use geom_col()

```
summer_usa %>% group_by(Medal) %>% summarise(count = n()) %>%
ggplot(aes(x = Medal, y = count)) + geom_col()
```

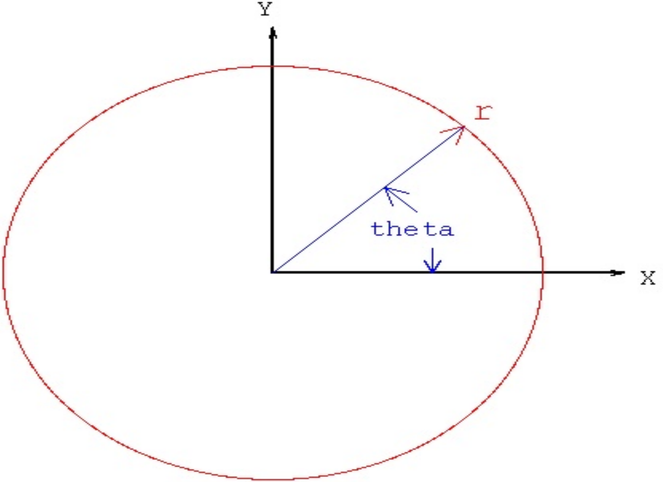


Univariate categorical variable

The polar coordinate system is a two-dimensional coordinate system in which each point on a plane is determined by a distance from a reference point and an angle from a reference direction.
parameters:

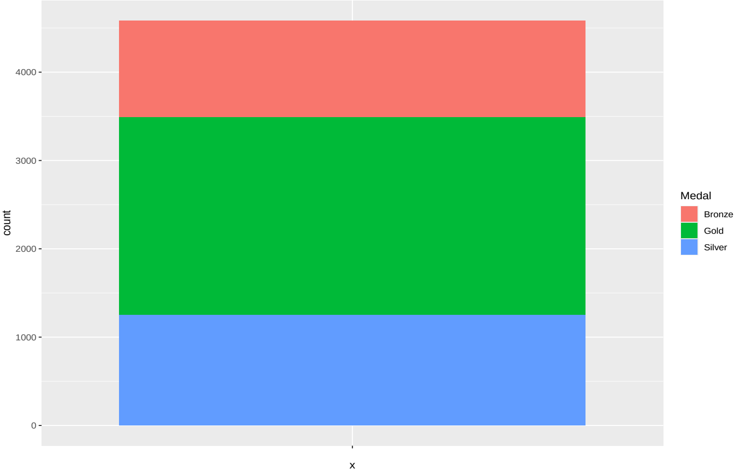
- θ - angle
- r radius

Univariate categorical variable



Univariate categorical variable

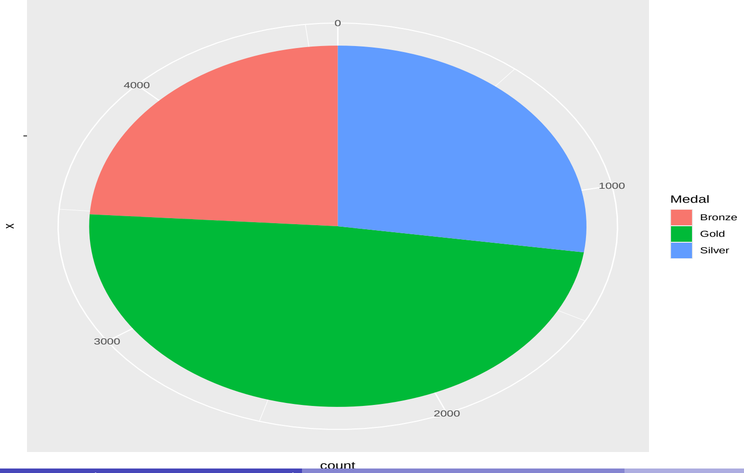
```
ggplot(data = summer_usa, aes(x = "", fill = Medal)) +
  geom_bar()
```



Univariate categorical variable

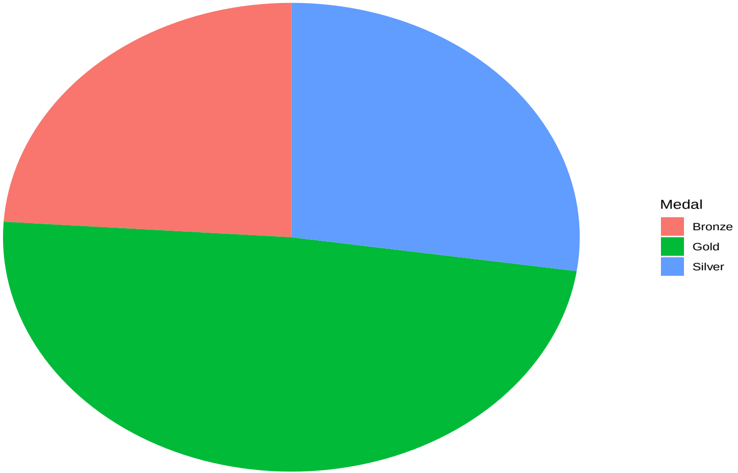
Counted y values are the θ parameter

```
ggplot(data = summer_usa, aes(x = "", fill = Medal)) +
  geom_bar() + coord_polar(theta = 'y')
```



Univariate categorical variable

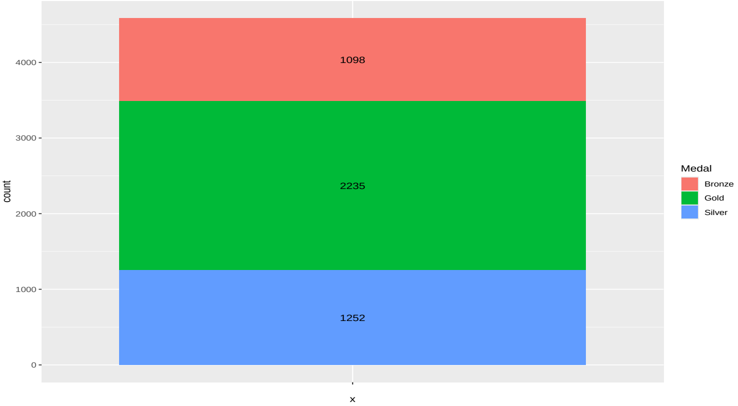
```
ggplot(data = summer_usa, aes(x = "", fill = Medal)) +
  geom_bar() + coord_polar(theta = 'y') + theme_void()
```



Univariate categorical variable

With pie charts it is also useful to add count values (relative or absolute) to the plot

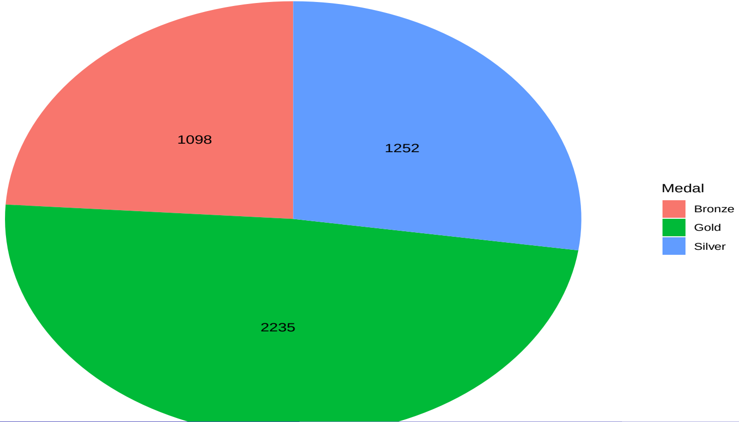
```
ggplot(data = summer_usa, aes(x = "", fill = Medal)) + geom_bar() +
    geom_text(stat = 'count', aes(label = ..count..),
              position = position_stack(vjust = 0.5))
```



Univariate categorical variable

To pie chart

```
ggplot(data = summer_usa, aes(x = "", fill = Medal)) + geom_bar() +
    geom_text(stat = 'count', aes(label = ..count..),
              position = position_stack(vjust=0.5)) +
  coord_polar(theta='y') + theme_void()
```



Univariate categorical variable

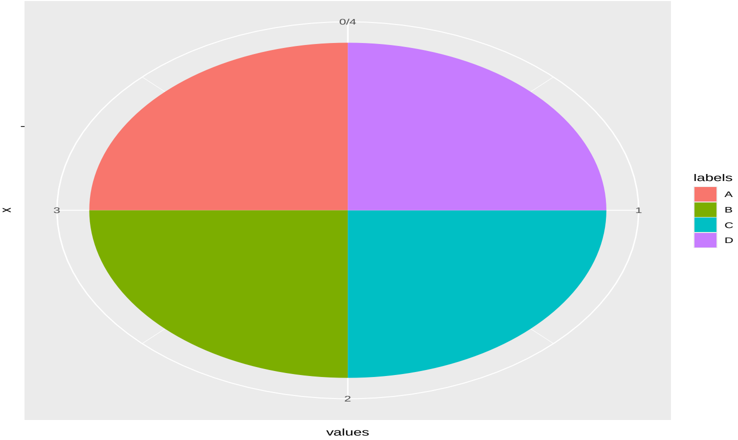
You can control the appearance of pie chart with the parameters:

- start - angle where the pie chart starts
- direction - do you want it to be clockwise or anti-clockwise

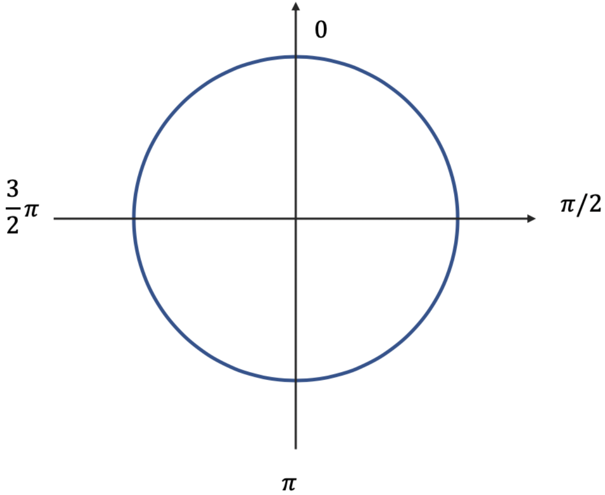
Univariate categorical variable

Understanding the **start** and **direction**

```
example <- data.frame(labels = LETTERS[1:4], values = rep(1,4))
ggplot(data = example, aes(x ="", y = values, fill = labels)) +
  geom_col() + coord_polar(theta = 'y')
```



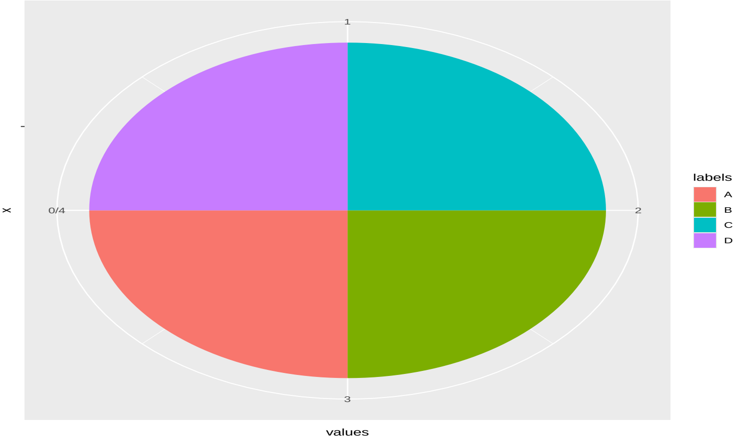
Univariate categorical variable



Univariate categorical variable

starting at $\frac{3}{2}\pi$

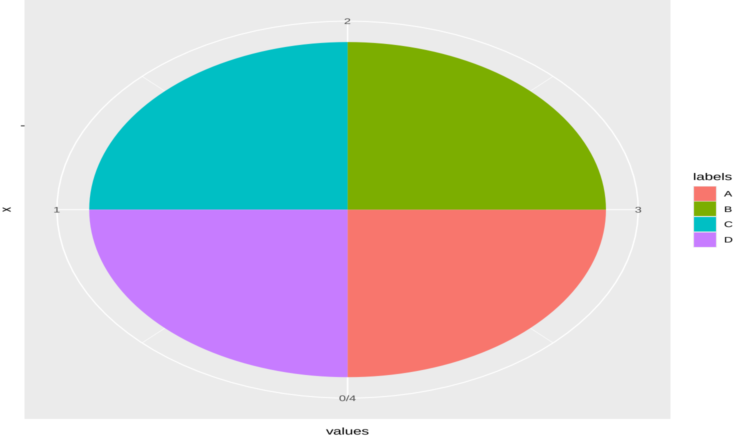
```
ggplot(data = example, aes(x ="", y = values, fill = labels)) +
  geom_col() + coord_polar(theta = 'y', start = 3*pi/2)
```



Univariate categorical variable

starting at π

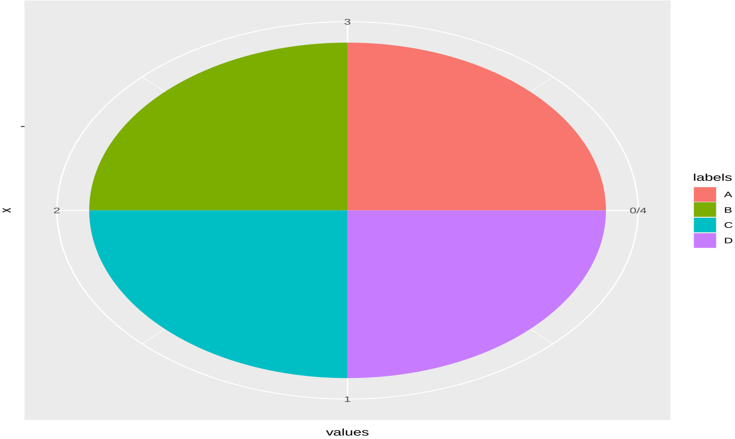
```
ggplot(data = example, aes(x ="", y = values, fill = labels)) +
  geom_col() + coord_polar(theta = 'y', start = pi)
```



Univariate categorical variable

starting at $\frac{\pi}{2}$

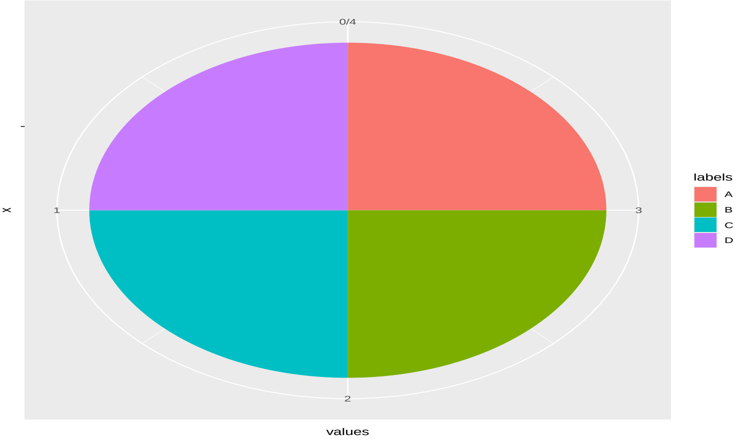
```
ggplot(data = example, aes(x ="", y = values, fill = labels)) +
  geom_col() + coord_polar(theta = 'y', start = pi/2)
```



Univariate categorical variable

Direction

```r
ggplot(data = example, aes(x ="", y = values, fill = labels)) + geom_col()
  coord_polar(theta = 'y', direction = -1)
```



Univariate categorical variable

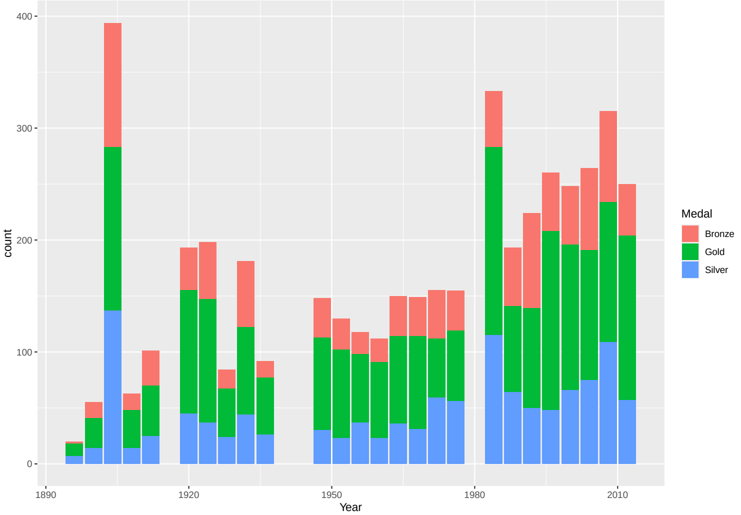
Don’t use pie charts !!

Multivariate categorical variable

- We can use barcharts to visualize multivariate categorical data as well.
- Look at the distribution of the medals over time (assuming year as categorical variable)
- by default the position is “stack”

Multivariate categorical variable

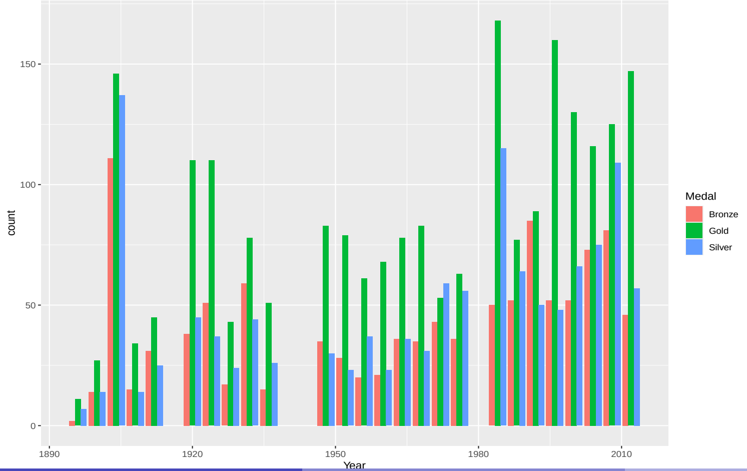
```
ggplot(summer_usa, aes(x = Year, fill = Medal)) + geom_bar()
```



Multivariate categorical variable

Position ‘dodge’ to get side-by-side bar chart

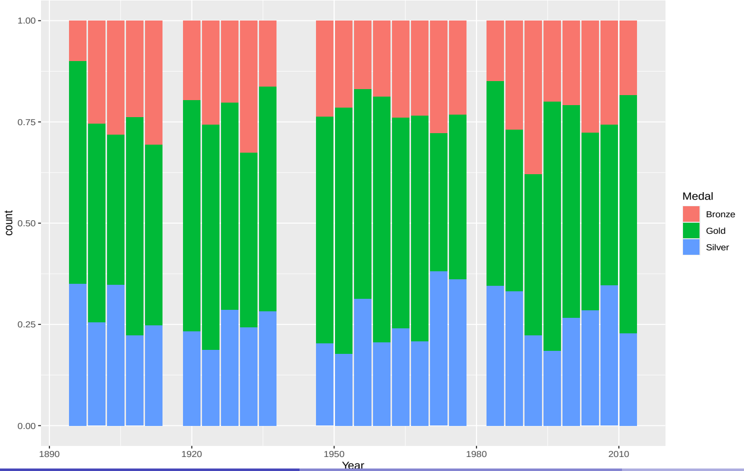
```
ggplot(summer_usa, aes(x = Year, fill = Medal)) +
  geom_bar(position = 'dodge')
```



Multivariate categorical variable

position = ‘fill’ to have stacked bar chart with the length of 1

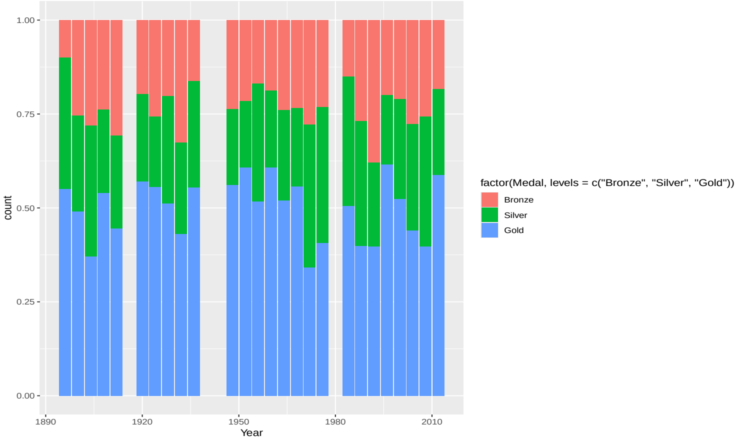
```
ggplot(summer_usa, aes(x = Year, fill = Medal)) +
  geom_bar(position = 'fill')
```



Multivariate categorical variable

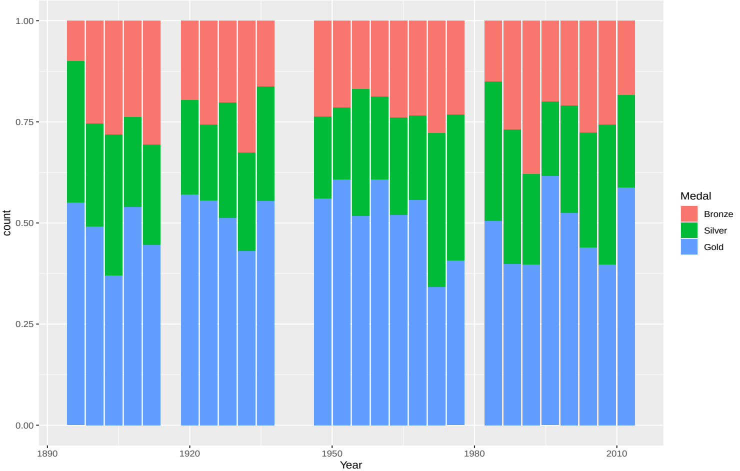
Change the order of variable categories

```
ggplot(summer_usa, aes(x = Year, fill = factor(Medal,
      levels=c("Bronze","Silver", "Gold" )))) +
  geom_bar(position = 'fill')
```



Multivariate categorical variable

```
ggplot(summer_usa, aes(x = Year, fill = factor(Medal,
      levels=c("Bronze","Silver", "Gold" )))) +
  geom_bar(position = 'fill') + labs(fill = 'Medal')
```



Multivariate categorical variable

- Bar charts are used to visualize contingency tables
- Useful to visualize conditional probabilities
- Relationship between variables

```r
load('Data/hr.rda')
colnames(hr)
##  [1] "satisfaction_level"    "last_evaluation"       "number_project"
##  [4] "average_montly_hours"  "time_spend_company"    "Work_accident"
##  [7] "promotion_last_5years" "sales"                 "salary"
## [10] "left"
```

Multivariate categorical variable

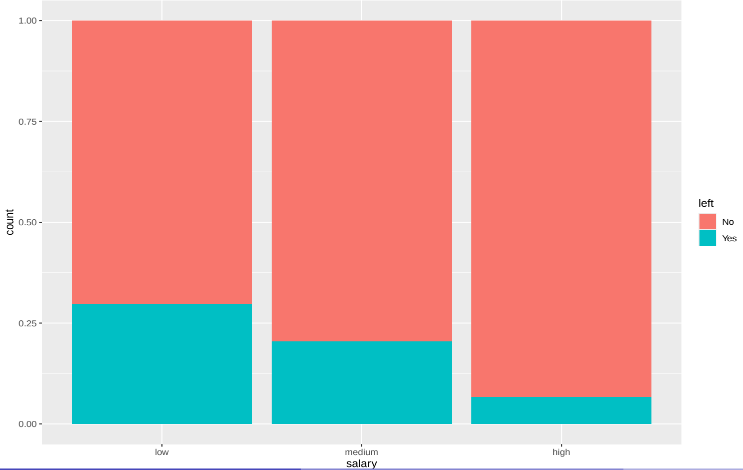
Why do people leave ?
Contingency table for conditional probabilities

```
prop.table(table(hr$salary, hr$left),1)
##
##                  No        Yes
##   low    0.70311646 0.29688354
##   medium 0.79568725 0.20431275
##   high   0.93371059 0.06628941
```

Multivariate categorical variable

Why do people leave ?

```
ggplot(hr, aes(x =salary, fill = left)) +
  geom_bar(position = 'fill')
```

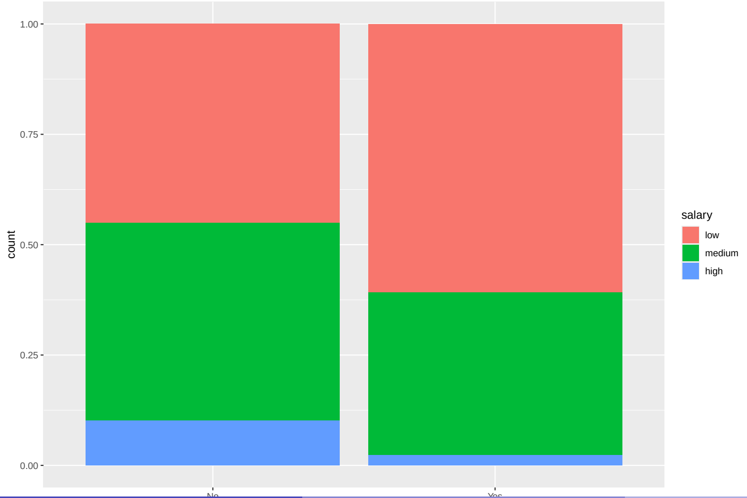


Multivariate categorical variable

```
prop.table(table(hr$salary, hr$left),2)
##
##                  No        Yes
##   low    0.45012251 0.60823299
##   medium 0.44880994 0.36880426
##   high   0.10106755 0.02296276
```

Multivariate categorical variable

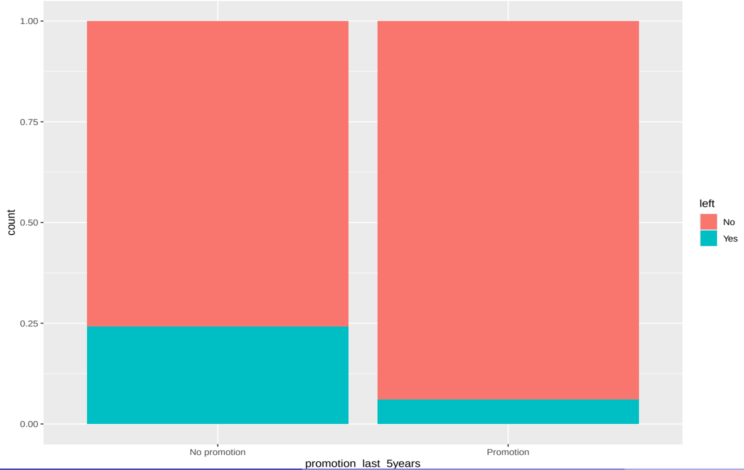
```r
ggplot(hr, aes(x = left, fill = salary)) +
  geom_bar(position = 'fill')
```



Multivariate categorical variable

Is promotion a factor ?

```r
ggplot(hr, aes(x = promotion_last_5years, fill = left)) +
  geom_bar(position = 'fill')
```



Case Study: Florence Nightingale

- Florence Nightingale was a British nurse during Crimean War in the middle of 19th century.
- She became very upset by a high mortality rate among British soldiers and started collecting data on the causes of deaths in hospitals.
- Based on her recommendations, sanitary conditions in hospitals were improved, which significantly decreased mortality rates in hospitals.

Case Study: Florence Nightingale

```
data("Nightingale")
str(Nightingale)
## 'data.frame':    24 obs. of  10 variables:
##  $ Date        : Date, format: "1854-04-01" "1854-05-01" ...
##  $ Month       : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 4 5 6 7 8
##  $ Year        : int  1854 1854 1854 1854 1854 1854 1854 1854 1854 1855
##  $ Army        : int  8571 23333 28333 28722 30246 30290 30643 29736 327
##  $ Disease     : int  1 12 11 359 828 788 503 844 1725 2761 ...
##  $ Wounds      : int  0 0 0 0 1 81 132 287 114 83 ...
##  $ Other       : int  5 9 6 23 30 70 128 106 131 324 ...
##  $ Disease.rate: num  1.4 6.2 4.7 150 328.5 ...
##  $ Wounds.rate : num  0 0 0 0 0.4 ...
##  $ Other.rate  : num  7 4.6 2.5 9.6 11.9 27.7 50.1 42.8 48 120 ...
```

Case Study: Florence Nightingale

To use the data in ggplot we need to transform it to long format first

```
nightingale_m <- melt(data = Nightingale, id.vars = 'Date',
                measure.vars = c('Disease', 'Wounds', 'Other'))
```

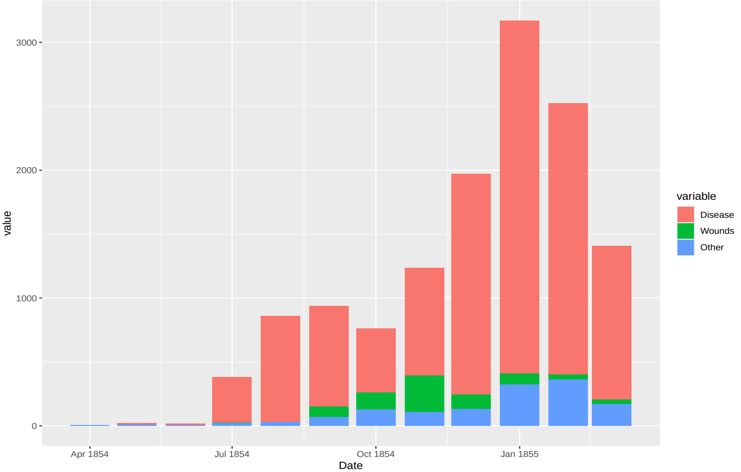
We will also create a dataset with the data for deaths before the sanitation was improved

```
night_before <- nightingale_m %>% filter(Date < '1855-04-01')
```

Case Study: Florence Nightingale

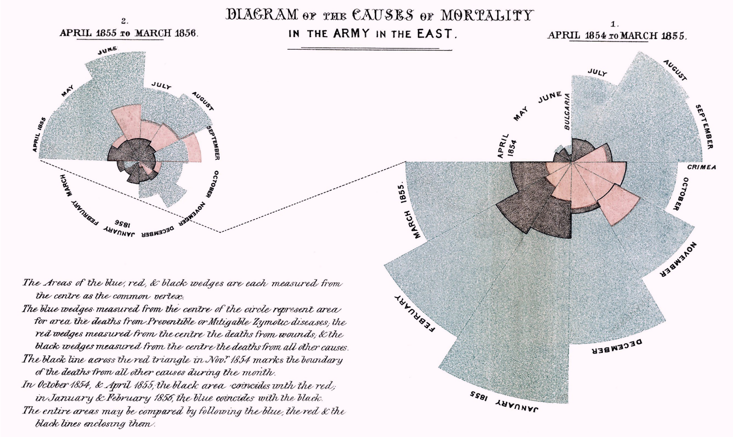
Subset the data for the early periods of Nightingale work (up to March 1855)

```
ggplot(night_before, aes(x = Date, fill = variable, y = value)) +
  geom_col()
```



Case Study: Florence Nightingale

The way how Florence Nightingale visualized this data, will become known as **Nightingale rose** or **coxcomb** chart



The Areas of the blue, red, & black wedges are each measured from the centre as the common vertex.

The blue wedges measured from the centre of the circle represent area for area the deaths from Preventible or Mitigable Zymotic diseases; the red wedges measured from the centre the deaths from wounds; & the black wedges measured from the centre the deaths from all other causes.

The black line across the red triangle in Nov.r 1854 marks the boundary of the deaths from all other causes during the month.

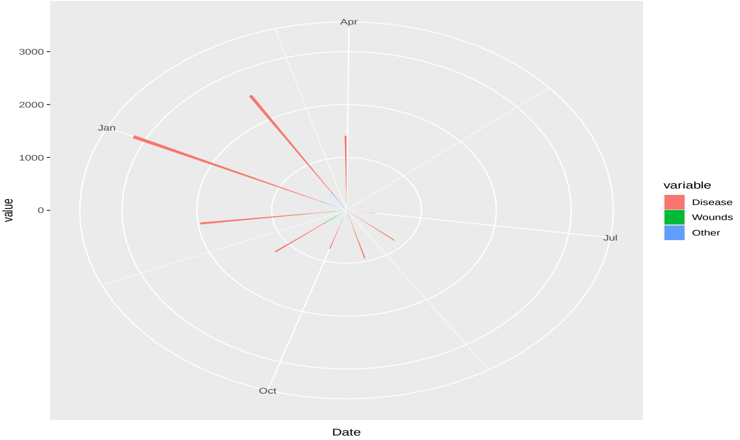
In October 1854, & April 1855, the black area coincides with the red; in January & February 1856, the blue coincides with the black.

The entire areas may be compared by following the blue, the red & the black lines enclosing them.

Case Study: Florence Nightingale

- We will try to replicate Nightingale's chart
- Pay attention, theta = 'x'

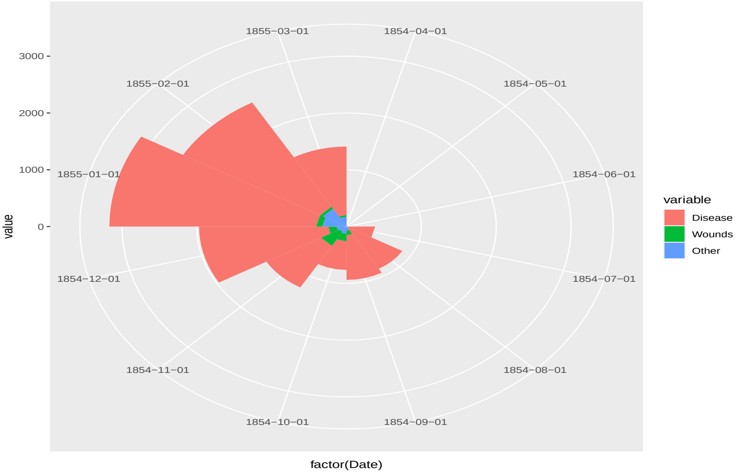
```
ggplot(night_before, aes(x= Date, fill = variable, y = value)) +
    geom_col(width = 1) + coord_polar(theta='x')
```



Case Study: Florence Nightingale

Date as factor, but still ugly

```r
ggplot(night_before, aes(x= factor(Date), fill = variable, y = value)) +
    geom_col(width = 1) + coord_polar(theta='x')
```



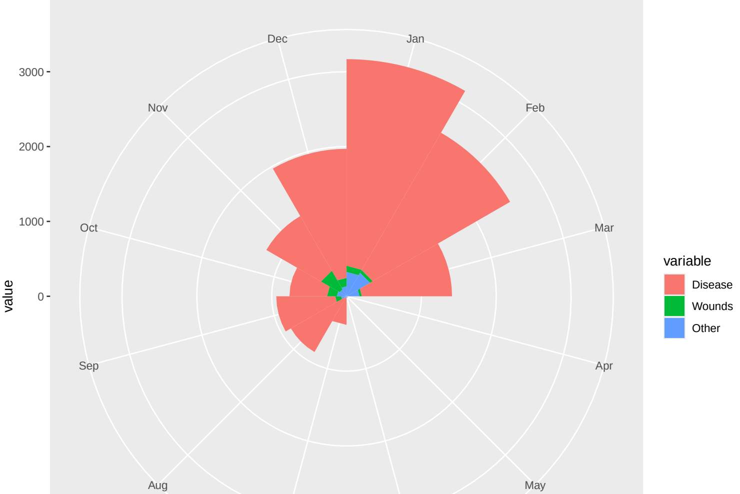
Case Study: Florence Nightingale

Make a Month variable as factor preserving the order of months

```r
night_before$Month <- factor(months(night_before$Date, abbreviate = T),
                             levels = month.abb)
```

Case Study: Florence Nightingale

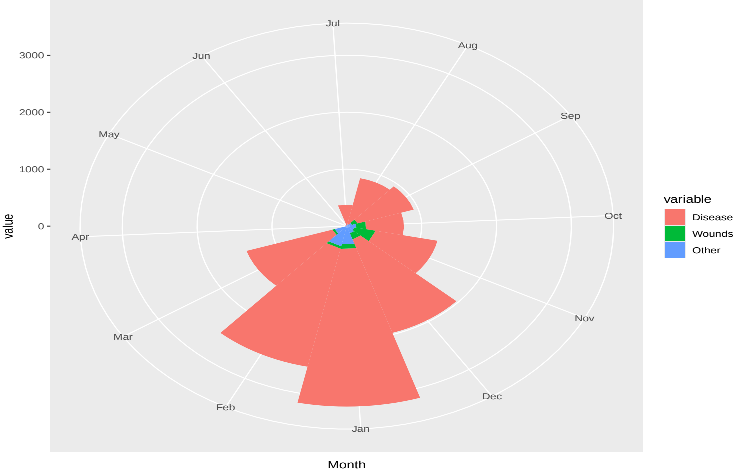
```
ggplot(night_before, aes(x=Month, fill=variable, y=value)) +
    geom_col(width = 1) + coord_polar(theta='x')
```



Case Study: Florence Nightingale

Set the starting point

```
ggplot(night_before, aes(x = Month, fill = variable, y = value)) +
    geom_col(width = 1) + coord_polar(theta='x', start=0.9*pi)
```

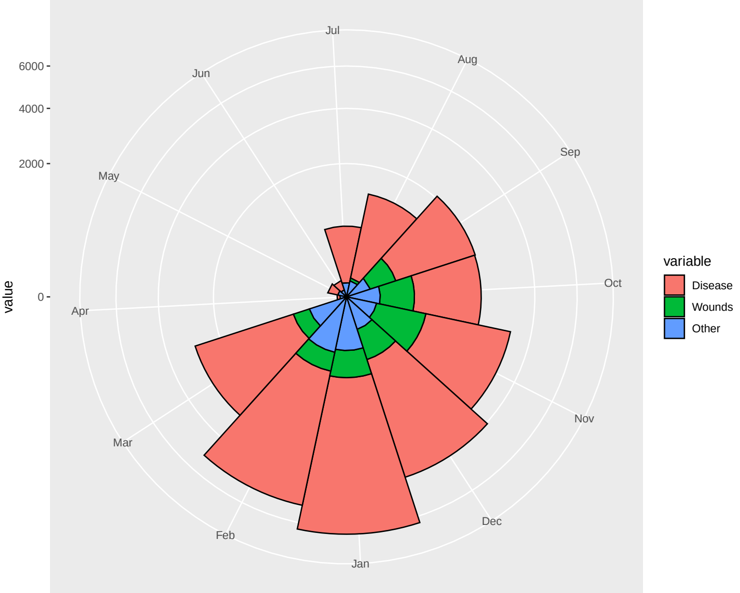


Case Study: Florence Nightingale

making better (color = ‘black’)

```r
ggplot(night_before, aes(x = Month, fill = variable, y = value)) +
  scale_y_sqrt() +
  geom_col(width = 1, color = 'black') +
  coord_polar(theta='x', start=0.9*pi)
```

Case Study: Florence Nightingale



Case Study: Florence Nightingale

Create a theme for the plot

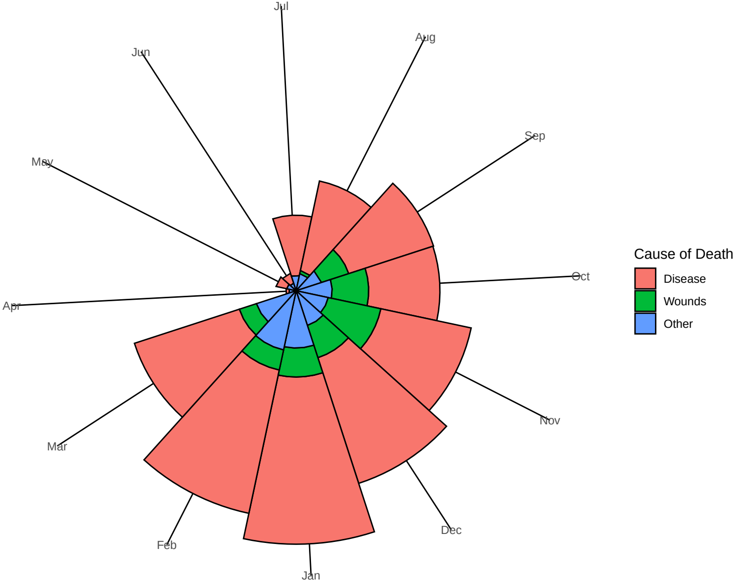
```
theme_coxcomb <-   theme(panel.background = element_blank(),
        axis.title = element_blank(),
        axis.text.y = element_blank(), axis.ticks.y = element_blank(),
        panel.grid.major.y = element_blank(),
        panel.grid = element_line(colour = 'black'))
```

Case Study: Florence Nightingale

Apply the theme

```r
ggplot(night_before, aes(x= Month, fill = variable, y = value)) +
  scale_y_sqrt() + geom_col(width = 1, color = 'black') +
  coord_polar(theta='x', start=0.9*pi) + labs(fill = 'Cause of Death') +
  theme_coxcomb
```

Case Study: Florence Nightingale



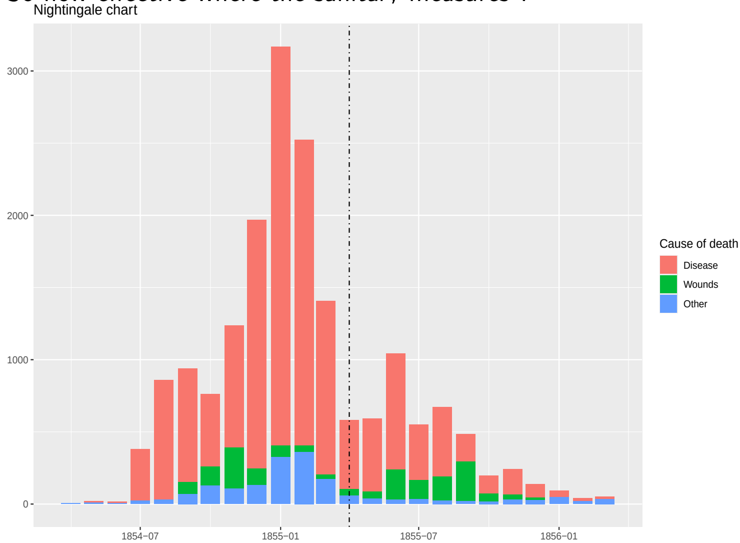
Case Study: Florence Nightingale

So how effective where the sanitary measures ?

```
ggplot(nightingale_m, aes(x = Date, fill = variable, y = value)) +
  geom_col() +
  geom_vline(xintercept=as.numeric(as.Date('1855-04-01')), linetype=4) +
  labs(title = "Nightingale chart", x = "", y = "",
       fill = "Cause of death")
```

Case Study: Florence Nightingale

So how effective where the sanitary measures ?


Nightingale chart
Cause of death
Disease
Wounds
Other