Data Visualization Project

Rory Quinlan

2022-10-12

Data Wrangling

- Exclude warnings in html output in r setup
- Load dplyr to access commands in package, such as the pipe command %>%
- Use the read.cvs() function to read the file and convert it to a data frame
- Assign this data frame a name (Dailyshowguests)
- Use the head() function to see the first 3 rows of the new dataset, to check it matches the cvs version
- Confirm that the new variable is a data frame with is.data.frame()

```
library(dplyr)
Dailyshowguests<- read.csv("C:/Users/roryq/Downloads/daily_show_guests.csv")
head(Dailyshowguests,3)
```

```
##   YEAR GoogleKnowlege_Occupation    Show  Group  Raw_Guest_List
## 1 1999                     actor 1/11/99 Acting  Michael J. Fox
## 2 1999                  Comedian 1/12/99 Comedy Sandra Bernhard
## 3 1999        television actress 1/13/99 Acting   Tracey Ullman
```

```
is.data.frame(Dailyshowguests)
```

```
## [1] TRUE
```

Data Wrangling

- Load tidyr to use tidr commands. specify dataset (Dailyshowguests), the one created above
- Then(%>%) group rows by year using the group_by() function
- Then(%>%) count the number of each group per year with count() function
- Then(%>%) create a new variable named Percent, with the mutate() function. Percent is defined by number of each group per year, divided by the total number of groups that year (sum(n)), total number of groups that year, is the sum of the number of each group per year.
- Assign this new data frame to new variable named Percent
- Use head() to see the first few row of data frame to check that the data frame is the format we would like

```
library(tidyr)
Percent <- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent = n / sum(n)*1
00)
Percent<- data.frame(Percent)
head(Percent,3)
```

```
##   YEAR  Group   n   Percent
## 1 1999 Acting 108 65.060241
## 2 1999 Comedy  25 15.060241
## 3 1999  Media  11  6.626506
```

Data Wrangling

Creating the Media Line to graph (Media)

- Create a new data frame same as previously done for Dailyshowguests, except add the filter() function to only show the group wanted (media)
- Assign this to a new variable (Media) to save it to be called on later
- Use the head() to view the first few rows, and check the code did what we intended

```
Media<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent=n / sum(n)*100) %
>% filter(Group== "Media")
year<- c(1999,2000,2001,2001,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015)
Media<-data.frame(year,Media)
```

Data Wrangling

Creating Action, Comedy, and Music Line to graph (AMCLine)

- Do the same, as we did for the Media Line, for each group whose data will be included on the graph in the Acting, Comedy, and Music line.
    - Select a specific group, set filter argument equal to that group
- Assign it to a new appropriate variable (A,C,M) to call on later
- Use head() function to test each new data frame

```
A<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent= n/sum(n)*100) %>% fi
lter(Group== "Acting")
head(A,3)
```

```
## # A tibble: 3 × 4
## # Groups:   YEAR [3]
##    YEAR Group      n Percent
##   <int> <chr>  <int>   <dbl>
## 1  1999 Acting   108    65.1
## 2  2000 Acting   100    59.2
## 3  2001 Acting    92    58.6
```

```
C<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent= n/sum(n)*100) %>% fi
lter(Group== "Comedy")
head(C,3)
```

```
## # A tibble: 3 × 4
## # Groups:   YEAR [3]
##    YEAR Group      n Percent
##   <int> <chr>  <int>   <dbl>
## 1  1999 Comedy    25   15.1
## 2  2000 Comedy    12    7.10
## 3  2001 Comedy    11    7.01
```

```
M<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent= n/sum(n)*100) %>% fi
lter(Group== "Musician")
head(M,3)
```

```
## # A tibble: 3 × 4
## # Groups:   YEAR [3]
##    YEAR Group        n Percent
##   <int> <chr>    <int>   <dbl>
## 1  1999 Musician    17   10.2
## 2  2000 Musician    13    7.69
## 3  2001 Musician    11    7.01
```

Data Wrangling

Creating Action, Comedy, and Music Line to graph (AMCLine)

Putting Together The Action, Comedy, and Music Line

- Create new variable(A1) to assign the new vector(created below) to call on later
    - Pull out the percent column of A to create a vector of the percent values for the Acting group for each year, by calling it with the $
    - repeat these steps above with the other groups in the Acting, Comedy, and music line (so C and M) are used to create C1 and M1 respectively
- Add these vectors (A1,M1,C1) together to add the three groups percents together
- Assign the added vectors to a new variable to call on later (AMC)

- We need to pair the percents of the AMC vector to years in order to graph the line
    - Create a vector called year that contains numeric string of all the years from the data set
    - Create the data frame that the Acting, Comedy, and Music line will be created from with the data.frame()function
        - Make each vector (AMC and year) a column in the arguments of the data.frame function, so a year is paired with the percent for that corresponding year.
    - Assign this data frame to a new variable and name it to graph later(AMCLine)
    - Use the head() function to view the first few rows, to check our work

```
A1<-A$Percent
M1<-M$Percent
C1<-C$Percent
AMC<-A1+C1+M1
AMC
```

```
##  [1] 90.36145 73.96450 72.61146 62.26415 56.02410 38.41463 37.03704 36.02484
##  [9] 25.53191 20.73171 20.85890 35.15152 33.74233 26.82927 42.77108 39.26380
## [17] 45.00000
```

```
year<- c(1999,2000,2001,2001,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015)
AMCLine<-data.frame(year,AMC)
head(AMCLine,3)
```

```
##   year      AMC
## 1 1999 90.36145
## 2 2000 73.96450
## 3 2001 72.61146
```

Data Wrangling

Creating Government and Politics Line to graph (GovLine)

- Create a new data frame same as Media was previously, except apply the filter() function to only show the group wanted (Government)
- Assign this to a new variable (G) to save it to be called on later
- Use the head() to view the first few rows, and check the code did what we intended

```
G<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent= n/sum(n)*100) %>% fi
lter(Group== "Government")
head(G,3)
```

```
## # A tibble: 3 × 4
## # Groups:   YEAR [3]
##    YEAR Group          n Percent
##   <int> <chr>      <int>   <dbl>
## 1  2001 Government     2   1.27
## 2  2002 Government     1   0.629
## 3  2003 Government     2   1.20
```

- Do the same for each group whose data will be included on the graph in the Government and Political line (Politician and Political Aide)
    - Select a specific group, apply filter to that group, and assign it to a new appropriate variable to call on later (P, PA)

```
P<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent= n/sum(n)*100) %>% fi
lter(Group== "Politician")
head(P,3)
```

```
## # A tibble: 3 × 4
## # Groups:   YEAR [3]
##    YEAR Group          n Percent
##   <int> <chr>      <int>   <dbl>
## 1  1999 Politician     2    1.20
## 2  2000 Politician    13    7.69
## 3  2001 Politician     3    1.91
```

```
PA<- Dailyshowguests %>% group_by(YEAR) %>% count(Group) %>% mutate(Percent= n/sum(n)*100) %>% f
ilter(Group== "Political Aide")
head(PA,3)
```

```
## # A tibble: 3 × 4
## # Groups:   YEAR [3]
##    YEAR Group              n Percent
##   <int> <chr>          <int>   <dbl>
## 1  2000 Political Aide     1   0.592
## 2  2001 Political Aide     1   0.637
## 3  2002 Political Aide     2   1.26
```

Data Wrangling

Creating Government and Politics Line to graph (GovLine)

Putting Together The Action, Comedy, and Music Line

- Create new variable(G1) to assign the new vector(created below) to call on later
- Pull out the percent column of G to create a vector of the percent values for the Government group for each year, by calling it with the $

    - repeat these steps above with the other groups in the Government and politics line (so P and PA) are used to create P1 and PA1 respectively
- Add these vectors (G1,P1,PA1) together to add the three groups percents together
- Assign the added vectors to a new variable to call on later (Gov)
- We need to pair the percents of the Gov vector to years in order to graph the line
    - Create the data frame that the Government and Politics line will be created from with the data.frame()function
        - Make each vector a column(year, Gov) in the arguments of the data.frame function, so a year is paired with the percent for that corresponding year.
            - year vector is previously defined, so we just call it back
    - Assign this data frame to a new variable and name it to graph later(GovLine)
    - Use the head() function to view the first few rows, to check our work

```
G1<-as.numeric(G$Percent)
P1<-as.numeric(P$Percent)
PA1<-as.numeric(PA$Percent)
Gov<-G1+P1+PA1
head(Gov,3)
```

```
## [1] 3.070421 8.958181 4.373509
```

```
year<- c(1999,2000,2001,2001,2003,2004,2005,2006,2007,2008,2009,2010,2011,2012,2013,2014,2015)
GovLine<-data.frame(year,Gov)
head(GovLine,3)
```

```
##   year      Gov
## 1 1999 3.070421
## 2 2000 8.958181
## 3 2001 4.373509
```

Data Visualization

Create Coordinate System and Specify Data Frame

- Load tidyverse to access ggplot commands
- Create a plot using the ggplot() function
    - Create a coordinate system for the plot using the mapping argument in the ggplot () function, and specify the x and y axis variables by setting them equal to to the x and y argument.

```
library(tidyverse)
plot<- ggplot(Percent,mapping=aes(x=year,y=Percent))
plot<-plot+geom_line(data=Media,mapping=aes(x=year,y=Percent), color= "darkorchid",size=1.5, na.
rm=TRUE)
```

Data Visualization

Adding Lines To the graph

Adding The Media Line To The Graph

- Add the first Line to the graph, by using Media data, adjust size, and color
  - Use the geom_line function to add a line graphing the Media data set on the plot
  - Specify Media data set in the data argument of geom_line
  - Specify the variables you want to graph on the plot (year and Percent) with the mapping, and x and y arguments
  - Specify the color of the line with the color argument of the geom_line function, and set it equal to desired color
  - Specify the size of the line with the size argument of the geom_line function, and set it equal to desired size

```
plot<-plot+geom_line(data=Media,mapping=aes(x=year,y=Percent), color= "darkorchid",size=1.5, na.
rm=TRUE)
```

Data Visualization

Adding Lines To the graph

Adding The Acting, Comedy, and Music Line To The Graph

- Use geom_line() function to add a line to the graph specify the data set as AMCLine from earlier with the data argument in the geom_line () function
  - Specify x and y variables in the AMCLine data frame that we want to graph (year, AMC) in the mapping and aes arguments, to graph percent values of the AMC column of the AMCLine dataset on the y axis. Graph year values from the year column of the data set on the x axis.
  - Specifying color, and size of the line with the color and size arguments in the geom_line function

```
plot<- plot+geom_line(data=AMCLine, mapping=aes(x=year,y=AMC),color="cornflowerblue",size=1.5, n
a.rm=TRUE)
```

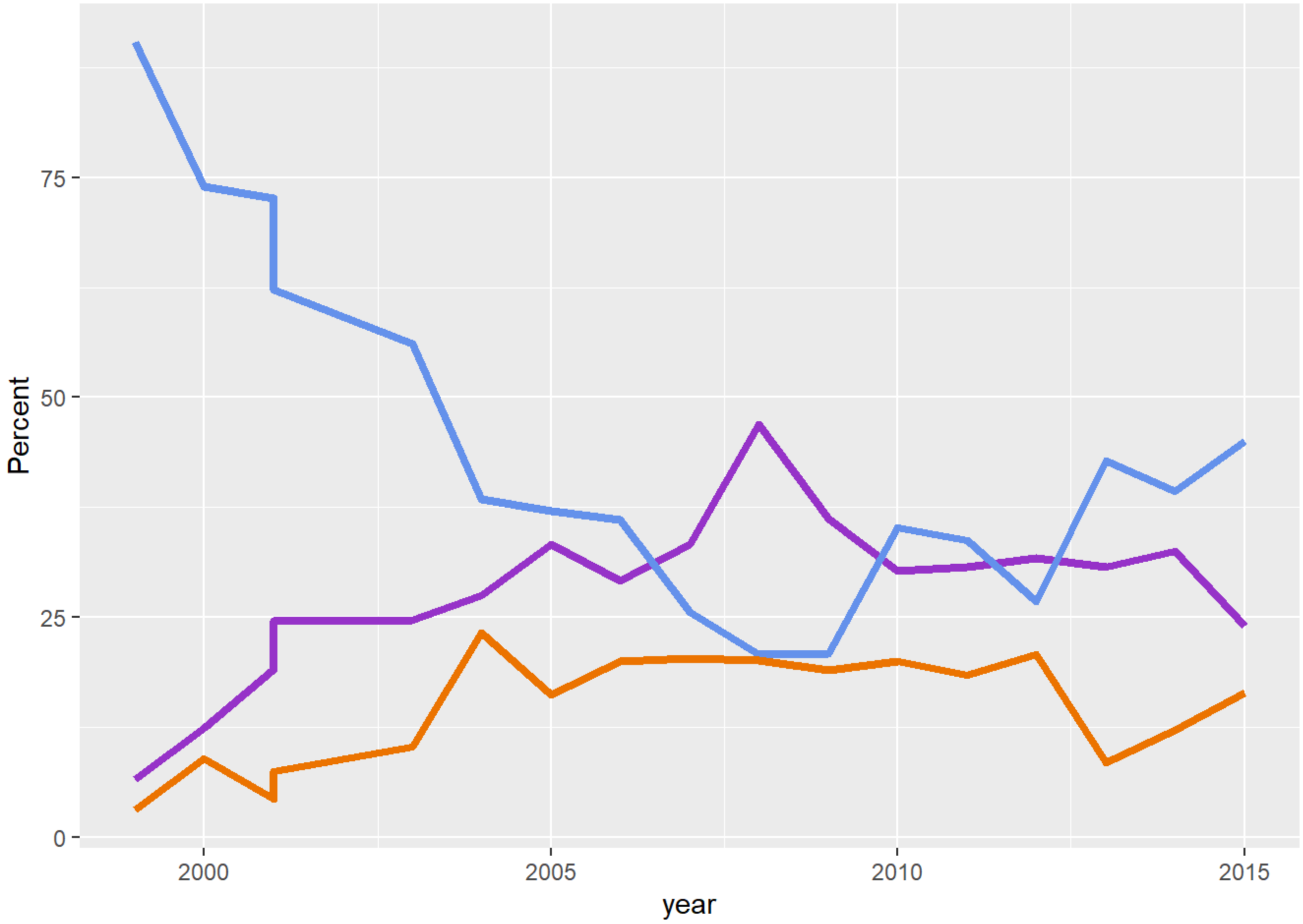
Data Visualization

Adding Lines To the graph

Adding The Government and Politics Line To The Graph

- Use geom_line() function to add a line to the graph specify the data set as GovLine from earlier with the data argument in the geom_line () function
    - Specify x and y variables in the GovLine data frame that we want to graph (year, Gov) in the mapping and aes arguments, to graph percent values in Gov column in data set on the y axis. Graph year values from the year column of the data set on the x axis.
    - Specifying color, and size of the line with the color and size arguments in the geom_line function

```
plot<-plot+geom_line(data=GovLine, mapping=aes(x=year,y=Gov), color="darkorange2", size=1.5)
plot
```



Data Visualization

Adding Context

- Manually adjust y scale from 0 to 100% to match the figure
    - Use the ylim() function and specify the minimum y value and maximum y value in the function

```
plot<-plot+ylim(0,100)
```

- Add Text labels to the lines, using the geom_text function

    - Specify x and y coordinates of the text with x and y arguments of the geom_text() function
    - Add the label argument to specify what the text will say
    - Add size and color arguments in the geom_text() function to match figure text color and size

```
plot<- plot+geom_text(x=2008,y=53, label="Media", colour="darkorchid",size=4.5,fontface="bold")
plot<- plot+ geom_text(x=2003.5,y=80, label="Acting, Comedy & Music", size=4.5, fontface="bold", color="cornflowerblue")
plot<- plot+geom_text(x=2012,y=5,label="Government and Politics", size=4.5, fontface="bold", color="darkorange2")
```

Data Visualization

Adding a Title and Subtitle To Graph

- Add a Title and subtitle with ggtitle() function
    - Add subtitle argument in ggtitle() function to add subtitle to match figure
    - Add theme()function
        - Add plot.title argument to specify that it is the title we are editing
            - plot.title argument is set equal to element text, because that it is an element in the graph made up of text. Specify that it is text and can be edited as such
        - Add face argument and set equal to bold in element_text() function, to change the title text to bold to match the figure
        - Add Size and family arguments to element_text() function to change text to match figure
    - Add another theme() function
        - Add plot.subtitle agrument set equal to element_text
        - Edit subtitle size, font, and face to match figure just the same as we edited the title

```
plot<-plot+ggtitle("Who Got To Be On 'The Daily Show'?", subtitle= "Occupation of guests, by year")
plot<-plot+ theme(plot.title=element_text(face="bold", size=15, family= "CM Roman"))
plot<-plot+theme(plot.subtitle=element_text(family="CM Roman",size=13))
```

Data Visualization

Edit axis labels

- Remove x and y auto generated labels to match figure.
    - Use xlab() and ylab() functions to specify the label of the x and y axis.
        - Add label argument to xlab() and ylab() functions.
            - Set label argument equal to NULL to eliminate plot labels.

```
plot<-plot+ylab(label=NULL)+xlab(label=NULL)
```

Data Visualization

Change x Axis Breaks and Labels

- Change breaks to match figure breaks name breaks to match figure breaks for the x axis.
    - Use the scale_x_continuous() function, and breaks argument to specify the breaks in the graph you want.
    - Use the scale_x_continuous() function and argument label to specify what the labels say on the graph.
        - We use scale_x_continuous, because x is a continuous variable.

```
plot<-plot + scale_x_continuous(breaks=c(2000, 2004, 2008,2012))
plot<-plot+ scale_x_continuous(labels=c("2000" = "2000", "2004" = "'04",
                              "2008" = "'08", "2012"="'12"))
```

Data Visualization

Change x Axis Breaks and Labels

- Adjust x axis text format to match the figure x labels, Use the theme() function with the argument for the x axis (axis.x.text.x), to adjust text on the x axis.
- Use the size and family argument within the x axis to adjust the size and font of the x axis labels.

```
plot<-plot + theme(axis.text.x = element_text(size=10, family="Serif", face="bold"))
```

Data Visualization

Change y Axis Breaks and Labels

- Adjust y axis text format, using the same code as above except changing the argument to axis.text.y to specify the y axis is the one we would like to edit.

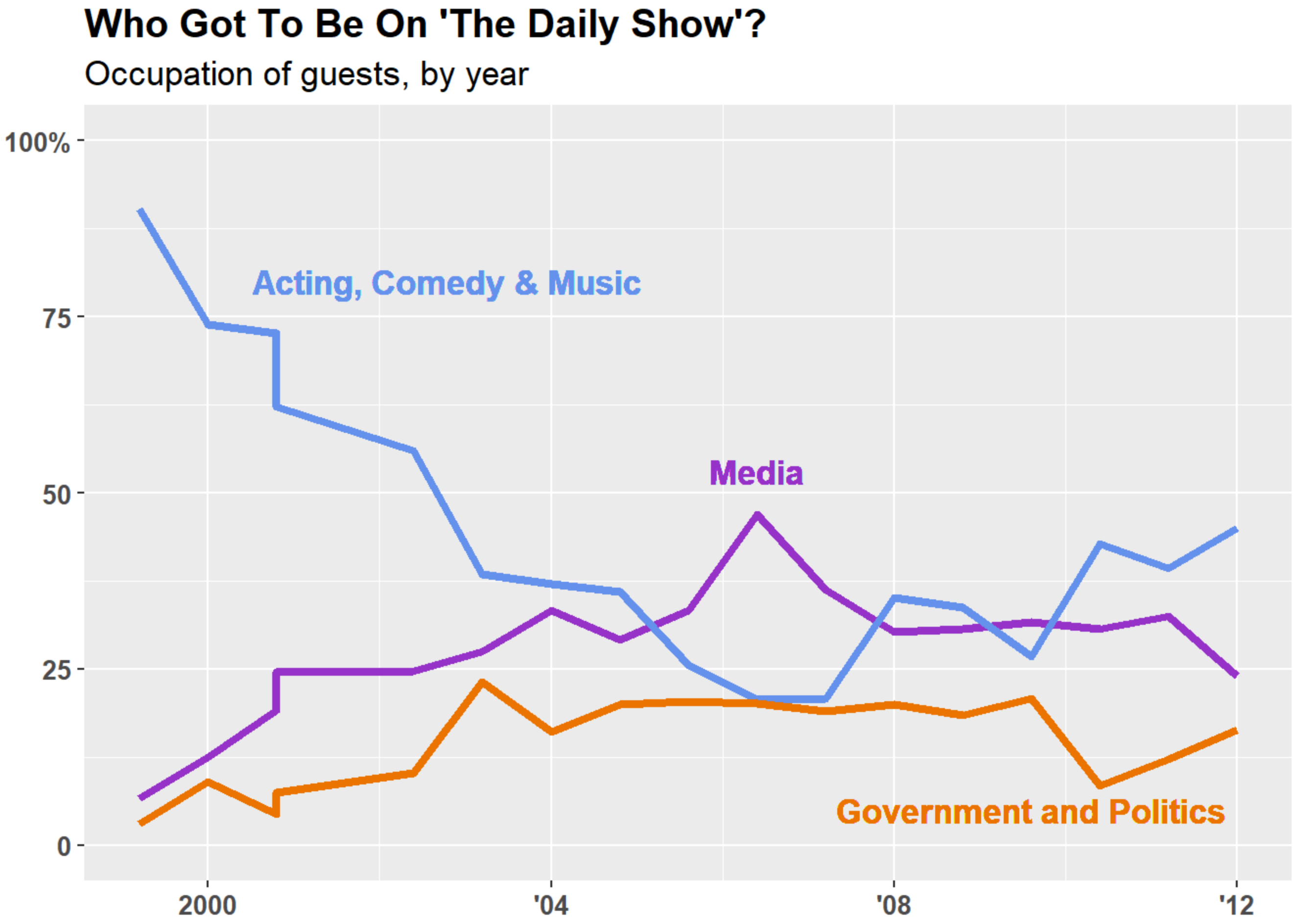
```
plot<-plot + theme(axis.text.y = element_text(size=10, family="Serif", face= "bold"))
```

- Label y axis breaks to match figure.
    - Note only the 100 on the figure has a percent sign on it.
- Use the same code as above for adjusting x axis labels except, change function to scale_y_continuous to adjust y axis labels.
- Adding the argument limit in the scale_y_continuous function to specify that the graph y axis lables should go from 0 to 100.

```
plot<-plot + scale_y_continuous(limit = c(0,100),
                     breaks = c(0,25,50, 75, 100),
                     labels = c("0","25","50","75", "100%"))
```

Recreated Figure

```
plot
```



Explanatory Paragraph

This graphic shows the percent composition of the guest types on the daily show per year. The guests are categorized into groups that relate to their occupation. On this graph they are grouped in one of three lines according to their occupation. Line one consists of guests in the acting, comedy, or music groups. Line two consists of guests in the media group, and line three consists of guests in the government or political groups. Groups are a variable from the original data set that categorizes their occupation. An example interpretation of this graph is that in the year 2000, 12.5% of guests on the daily show had a occupation related to media. In that same year (2000), 75% of guests on the daily had an occupation in acting, comedy, or music.

Original Article

Every Guest That Jon Stewart Ever Had On “The Daily Show”

Original Article Link (http://fivethirtyeight.com/datalab/every-guest-jon-stewart-ever-had-on-the-daily-show/)