STAT/CS 187 Homework 3

Solutions

Spring 2022

Set Up Your Project and Load Libraries

```
knitr::opts_chunk$set(echo = TRUE)

## Set the default size of figures
knitr::opts_chunk$set(fig.width=8, fig.height=5)


## Load the libraries we will be using
library(readr)
library(dplyr)
library(tidyverse)
library(ggplot2)
library(lattice)
library(directlabels)
```

```
## Warning: package 'directlabels' was built under R version 4.1.3
```

Question 1: Board Game Geek Reviews by Category

See the word doc for the description of the data.

```
# Read in the board game data below:
bgs <-   read_csv("C:/Users/huaye/Desktop/CS 187A/HW/HW_3/board games.csv")
```

```
## Rows: 20351 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): name, game_category, rating_group
## dbl (4): year_published, difficulty, rating_avg, rating_sd
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(bgs)

# Next, remove any games with the game_category "Other" and name it bg1
bg1 <- bgs %>%
  filter(game_category !="Other")

View(bg1)


# Then change the rating_group to a factor with ORDERED levels Terrible, Bad, Average, Good, Great
```

```
rank<-as.factor(bg1$rating_group)
bg1 <-
  bg1 %>% mutate(rating_group = factor(rank, ordered = TRUE, levels = c("Terrible","Bad","Average","Good


# Use the line below to check that rating is a ordered factor
str(bg1)
```

```
## tibble [9,167 x 7] (S3: tbl_df/tbl/data.frame)
##  $ name          : chr [1:9167] "Shifti" "Archimedes" "Checkers" "Top Hats" ...
##  $ year_published: num [1:9167] 1977 1981 1150 1997 1987 ...
##  $ difficulty    : num [1:9167] 1 2.17 1.76 1.17 2.33 ...
##  $ rating_avg    : num [1:9167] 4.7 4.85 4.89 4.59 4.2 ...
##  $ rating_sd     : num [1:9167] 1.66 1.28 1.53 1.4 1.73 ...
##  $ game_category : chr [1:9167] "Abstract" "Abstract" "Abstract" "Abstract" ...
##  $ rating_group  : Ord.factor w/ 5 levels "Terrible"<"Bad"<..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
# Create the side-by-side bar chart for rating by category:
bg1 %>% group_by(game_category)
```

Part 1a) Side-by-Side Bar Chart

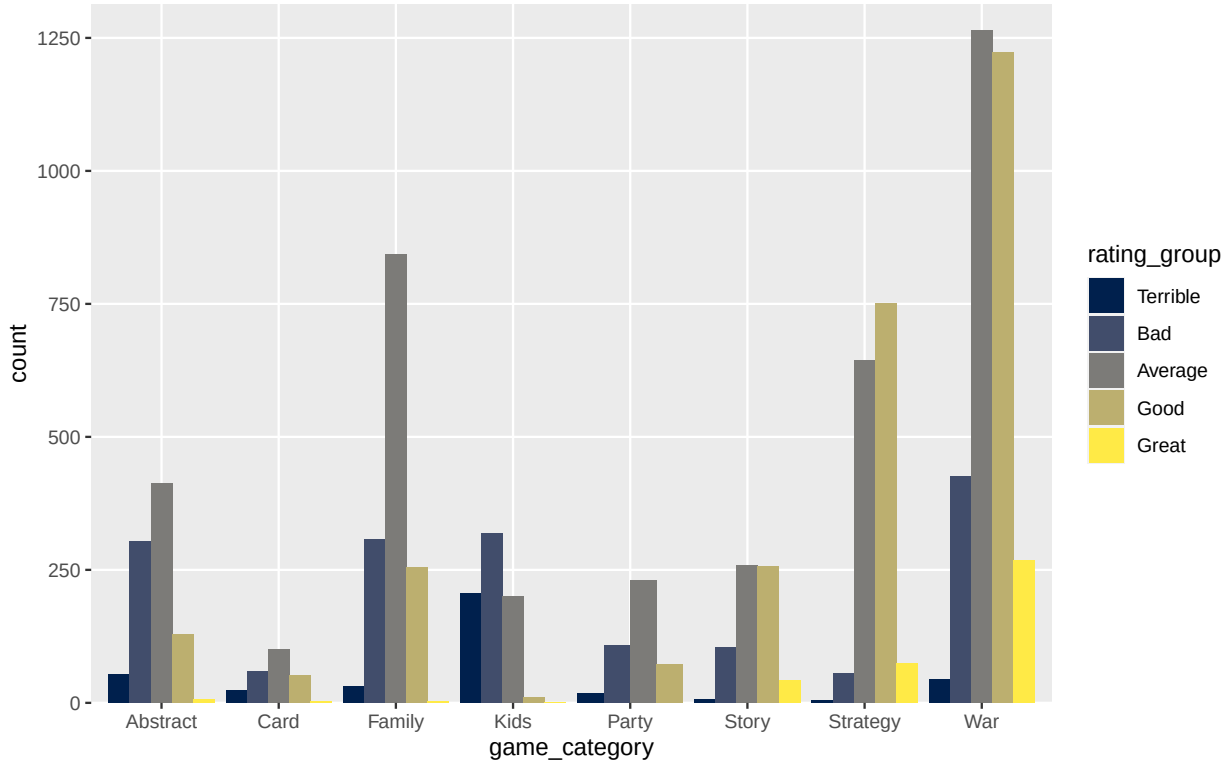
```
## # A tibble: 9,167 x 7
## # Groups:   game_category [8]
##    name              year_published difficulty rating_avg rating_sd game_category
##    <chr>                      <dbl>      <dbl>      <dbl>     <dbl> <chr>
##  1 Shifti                      1977       1          4.70      1.66 Abstract
##  2 Archimedes                  1981       2.17       4.85      1.28 Abstract
##  3 Checkers                    1150       1.76       4.89      1.53 Abstract
##  4 Top Hats                    1997       1.17       4.59      1.40 Abstract
##  5 Eye                         1987       2.33       4.20      1.73 Abstract
##  6 UNO Dice                    1987       1.13       4.87      1.64 Abstract
##  7 Hack Attack                 2000       1.5        4.12      1.74 Abstract
##  8 British Square              1978       1.25       4.62      1.72 Abstract
##  9 Doctor Who: Bat~            1989       1          4.55      1.79 Abstract
## 10 UNO Dominos                 1986       1.5        4.81      1.66 Abstract
## # ... with 9,157 more rows, and 1 more variable: rating_group <ord>
```

```
  ggplot(data=bg1, mapping=aes(x=game_category,fill=rating_group))+

  geom_bar(position = position_dodge())+

# Add the code below to you gg_object for the graph to match what is in the homework instructions
  scale_y_continuous(minor_breaks = NULL,
                     expand = expansion(mult = 0,
                                        add = c(0, 50))) +

  scale_fill_viridis_d(option = "cividis")
```




```
# Create a data set that has the rating proportions for each individual game_category

data1<-bgs %>%
  filter(game_category !="Other") %>%
  group_by(game_category,rating_group) %>%
  summarise(n_group=n()) %>%
  mutate(proportion = n_group/sum(n_group))
```

Part 1b) Conditional Proportions - Rating by Category

```
## `summarise()` has grouped output by 'game_category'. You can override using the
## `.groups` argument.
```

```
rating_by_category <- data.frame(data1$game_category,data1$rating_group,data1$proportion)

rank<-as.factor(rating_by_category$data1.rating_group)
rating_by_category <- rating_by_category %>%
  mutate(data1.rating_group = factor(rank, ordered = TRUE, levels = c("Terrible","Bad","Average","Good"


# Run to check that the data.frame is cone correctly
tibble(rating_by_category)
```

```
## # A tibble: 39 x 3
##    data1.game_category data1.rating_group data1.proportion
##    <chr>               <ord>                         <dbl>
##  1 Abstract            Average                       0.455
##  2 Abstract            Bad                           0.336
##  3 Abstract            Good                          0.143
```

```
##  4 Abstract           Great                0.00773
##  5 Abstract           Terrible             0.0586
##  6 Card               Average              0.423
##  7 Card               Bad                  0.247
##  8 Card               Good                 0.218
##  9 Card               Great                0.0126
## 10 Card               Terrible             0.100
## # ... with 29 more rows
```

Part c) Relative Side-by-Side Bar Chart Using the data.frame created in part b), create a plot similar to the barchart in part a), but with the proportion on the y-axis instead of the count

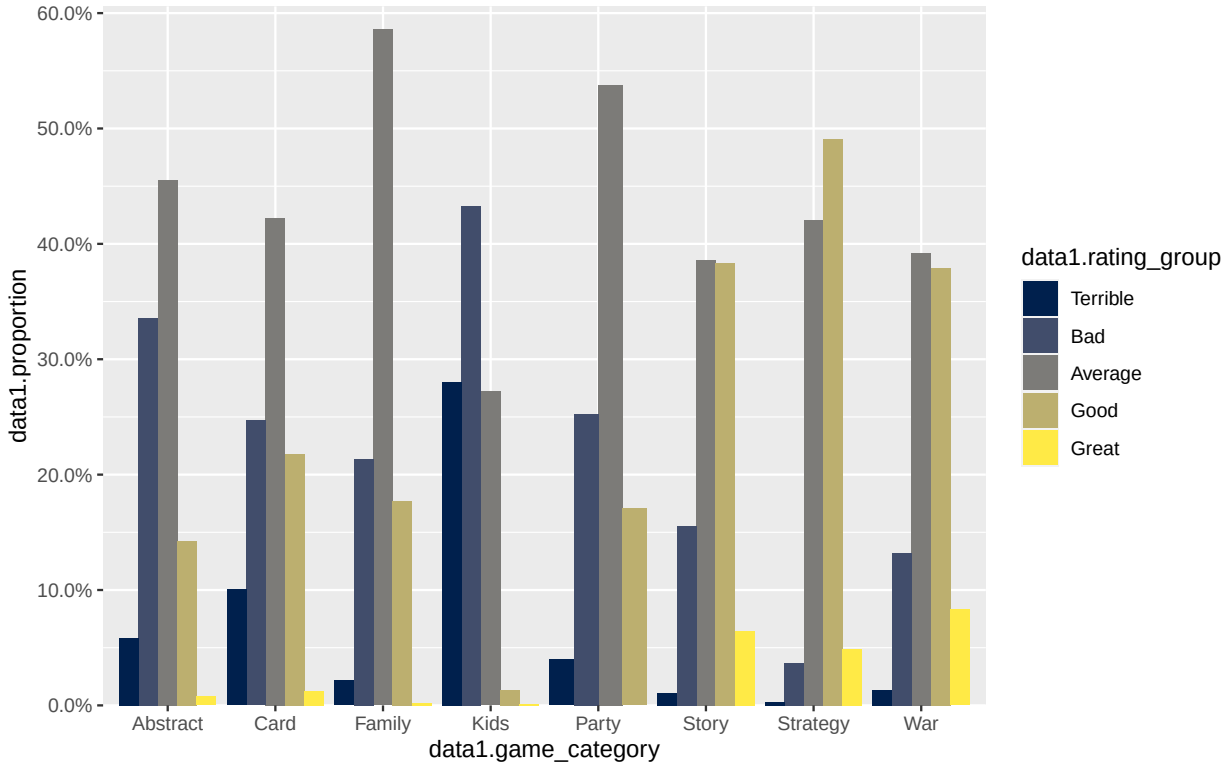
```
# Create the Side-by-Side box plot shown in the homework instructions:

  ggplot(data=rating_by_category,mapping=aes(x=data1.game_category,y=data1.proportion,fill=data1.rating_

  geom_col(position = position_dodge())+


# Similar to 1a), add this code for your graph to match the homework description
  scale_y_continuous(labels = scales::percent,
                     expand = expansion(mult = 0,
                                        add = c(0, 0.020)))  +

  scale_fill_viridis_d(option = "cividis")
```



Part d) Comparing Game Categories War games tend to have higher ratings (at least by people who use BoardGameGeek.com).

The two graphs in a) and c) look different because the y-axis aredifferent(y-axis for graph a) is count, but y-axis on graph c) is proportion).

Question 2: More Board Games - Rating by Year

Question 2 will use the full data set (all 20,000+ board games), not the smaller set from Question 1

Part 2a) Average Rating by Year Create a data set named *games_cat_year* that summarizes the data for each year_published and game_category combination. The data set should have the following columns:

- category = The game_category
- year = The publishing year
- games = the number of games for the category and year combination
- rating_mean = the average rating for all games of the category released for the year
- rating_sdev = the standard deviation for rating_avg for all games of the category released for the year

Only keep the rows from 1990 to the present

```
# Create the new data.frame
games_cat_year1 <- bgs %>%
  select(game_category,year_published,rating_avg,rating_sd)  %>%
  rename(category=game_category,year= year_published,rating_mean=rating_avg,rating_sdev= rating_sd) %>%
  filter(year> 1990 | year == 1990) %>%
  group_by(category,year)



games_cat_year2 <- games_cat_year1 %>%
  filter(year> 1990 | year == 1990) %>%
  group_by(category,year) %>%
  summarise(n_game=n()) %>%
  mutate(games=n_game)
```

```
## `summarise()` has grouped output by 'category'. You can override using the
## `.groups` argument.
```

```
games_cat_year <- full_join(games_cat_year1,games_cat_year2)%>%
  select(-n_game)#Join dataset and drop column n_game
```

```
## Joining, by = c("category", "year")
```

```
# Use the code below to print your results. Should match the results in the homework description
games_cat_year %>%

  group_by(category) %>%

  slice_max(n = 1,
            order_by = games)
```

```
## # A tibble: 1,507 x 5
## # Groups:   category [9]
##    category  year rating_mean rating_sdev games
##    <chr>    <dbl>       <dbl>       <dbl> <int>
##  1 Abstract  2006        4.69       1.61     48
##  2 Abstract  2006        4.62       1.89     48
##  3 Abstract  2006        5.71       1.61     48
##  4 Abstract  2006        5.95       0.833    48
```

```
##  5 Abstract  2006        5.97       1.58     48
##  6 Abstract  2006        5.32       1.57     48
##  7 Abstract  2006        5.65       1.81     48
##  8 Abstract  2006        5.91       1.38     48
##  9 Abstract  2006        5.48       1.83     48
## 10 Abstract  2006        5.58       1.50     48
## # ... with 1,497 more rows
```

Part 2b) Linegraph for Rating by Year per Category Create a line graph of rating_mean by year only for Strategy games. Make sure that the plot background and labeling looks the same. The line color is "orchid" and has size 1

```
# Create and save the line graph
plot_dat <- games_cat_year %>%
  filter(category %in% c("Strategy")) %>%
  group_by(year,games)

gg_strategy_rating <- ggplot(data=plot_dat , mapping = aes(x=year,y=rating_mean))+
  geom_line(color="orchid",size=1)+
  xlab("Year")+
  ylab("Average Rating")+
  scale_x_continuous(limits = c(1990, 2020),
                     breaks = seq(1990, 2020, by = 10),
                     minor_breaks = NULL)+
  scale_y_continuous(limits = c(6,7.5),
                     breaks = seq(6, 7.5, by = 0.5),
                     minor_breaks = NULL)


  distinct(plot_dat,year)
```

```
## # A tibble: 32 x 2
## # Groups:   year, games [32]
##     year games
##    <dbl> <int>
##  1  1997     9
##  2  1993     5
##  3  2003    30
##  4  2012    91
##  5  1999    22
##  6  1991     8
##  7  1998    13
##  8  2001    20
##  9  2002    27
## 10  2004    38
## # ... with 22 more rows
```

```
# Print the graph in the knitted document
gg_strategy_rating
```

Part 2c) Smoothed Line for Overall Mean by Year Add a smoothed LOESS line to the previous graph for the average rating for games per year, not separated by category. Set the following aesthetics for the smoothed line:

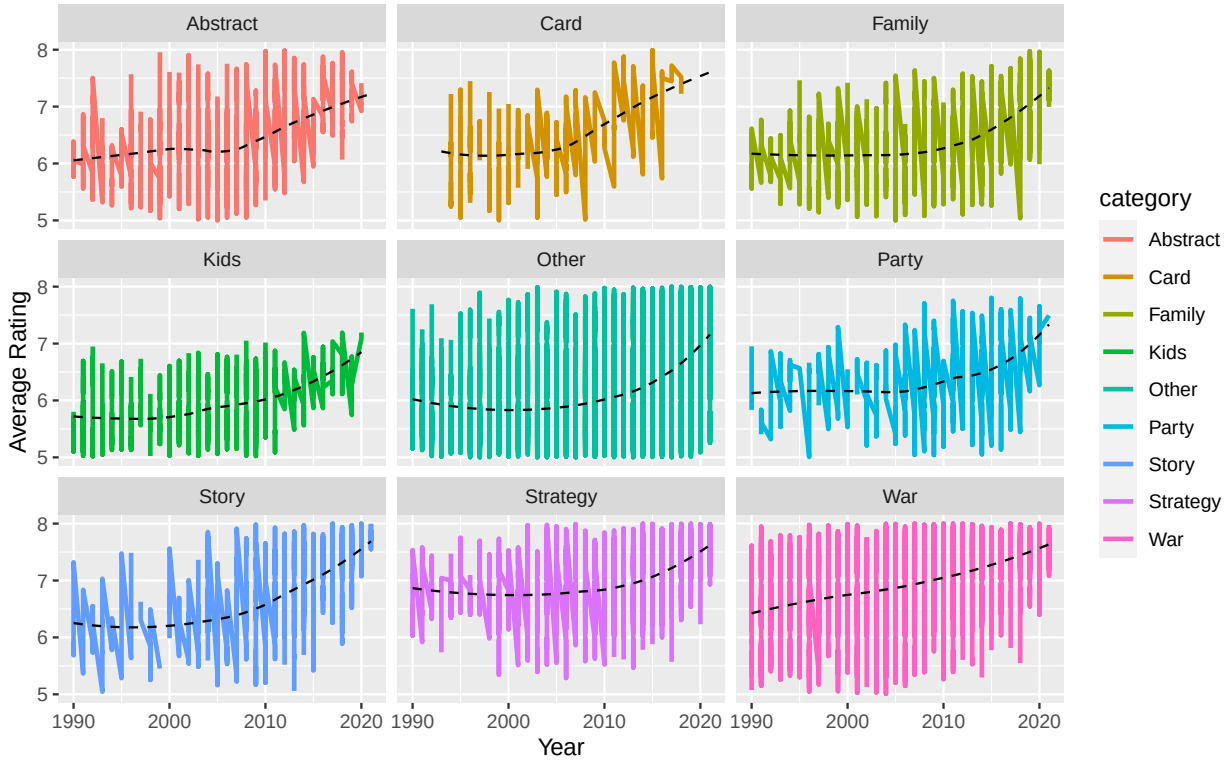
- size = 0.5
- color = "black"
- linetype = "dashed"

```
# Build on the plot created in 2b)
gg_strategy_rating + geom_smooth(method="loess",size = 0.5, color = "black", linetype = "dashed",se=FALS
```

Part 2d) Each Game Category Use small multiples to create the same graph from 2c) for all 9 game categories.

What can you determine about the opinions of Board Game Geek users about the different category of games?

```
# Create the small multiple plots seen in the homework description
ggplot(data=games_cat_year,mapping =aes(x=year,y=rating_mean,color=category))+
  geom_line(size=1)+
  ylim(5,8)+
  geom_smooth(method="loess",size = 0.5, color = "black", linetype = "dashed",se=FALSE)+
  facet_wrap(~category,nrow=3)+
  xlab("Year")+
  ylab("Average Rating")
```



Question 3: Word Data

```
# Read the word data into the R environment
words <- read_csv("C:/Users/huaye/Desktop/CS 187A/HW/HW_3/words.csv")
```

Part a) Change word level order

```
## Rows: 688 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): word, word_type
## dbl (2): Year, relative_usage
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Change word to a factor with the level order as idiot/moron/nimrod/nerd/geek/dork
rank<-as.factor(words$word)
words <-  words %>%
  mutate(word = factor(rank, ordered = TRUE, levels = c("idiot","moron","nimrod","nerd","geek","dork")))


# Use the code below to check that the words are in the correct order
levels(words$word)
```

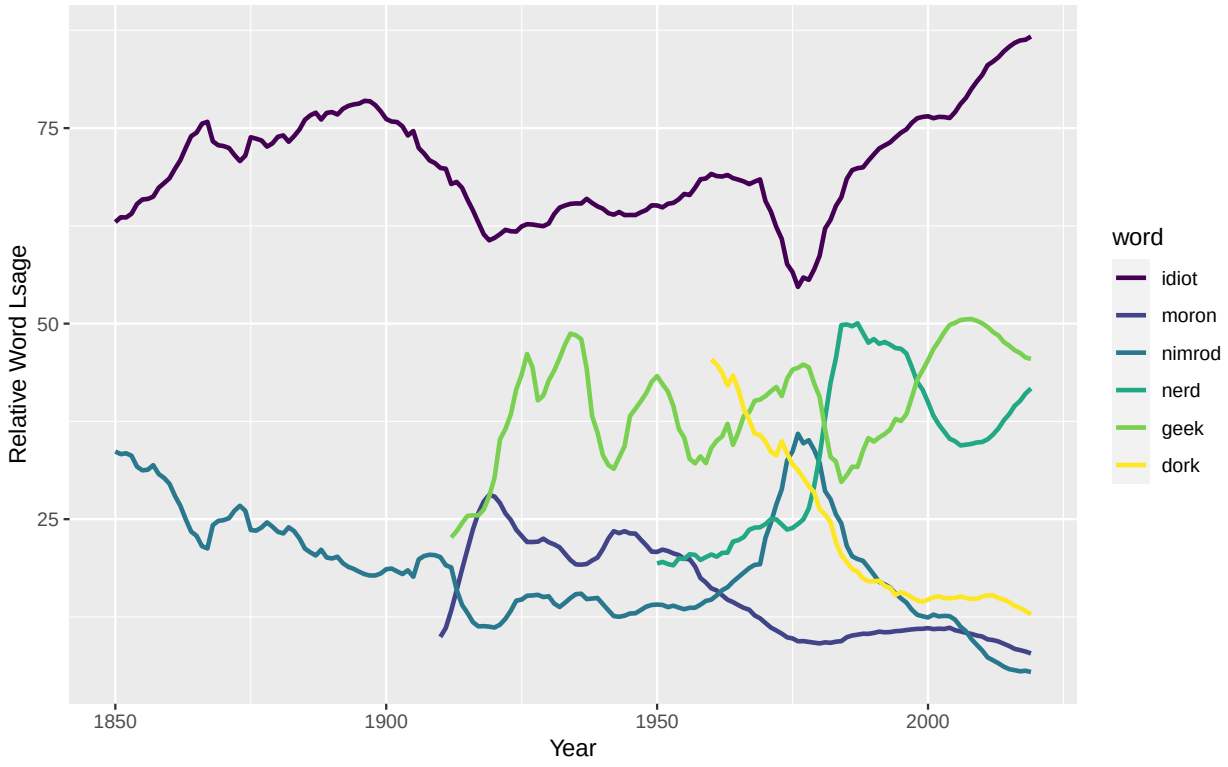
```
## [1] "idiot"  "moron"  "nimrod" "nerd"   "geek"   "dork"
```

3b) Graph of words over time Create the line graph below for relative usage by year. The size of each line is 1

```
# Create and save a plot of relative usage by year for each word
gg_word_line <- ggplot(data=words,mapping =aes(x=Year,y=relative_usage,,color=word))+
  geom_line(size=1)+
  xlab("Year")+
  ylab("Relative Word Lsage")



# Print the plot in the knitted word document
gg_word_line
```



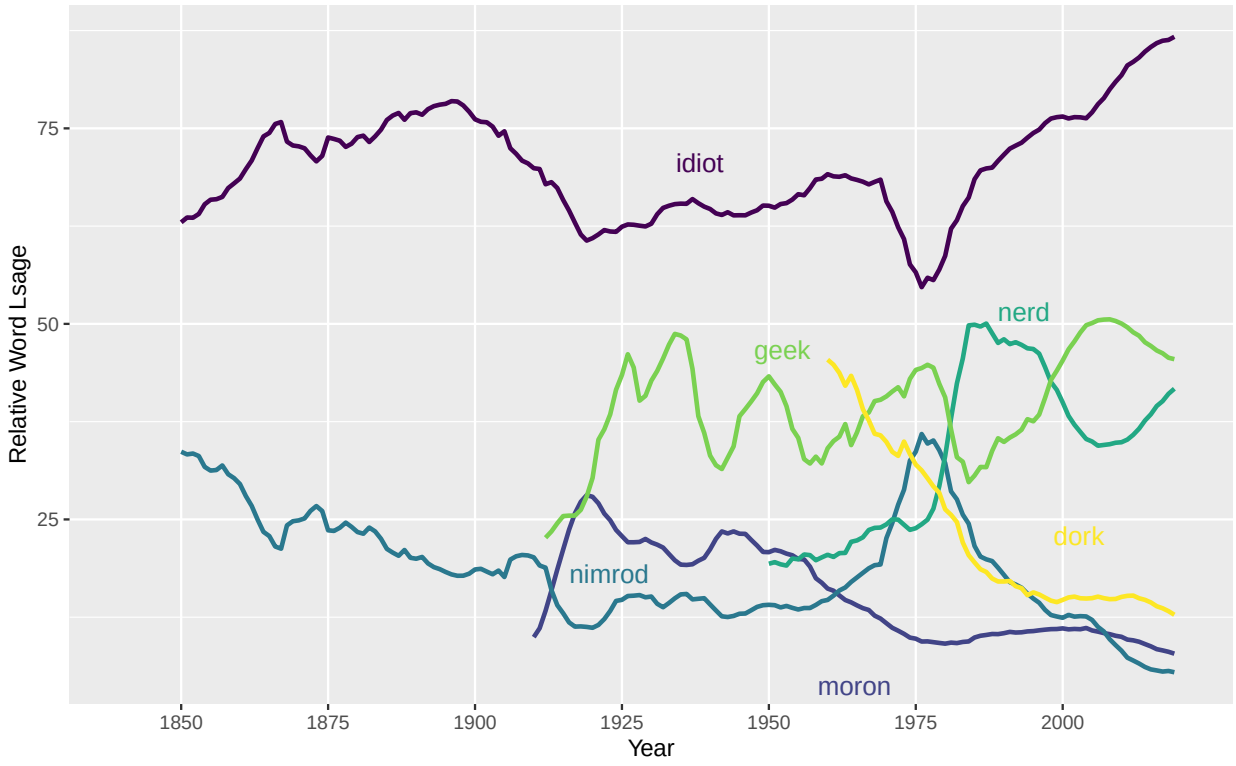
3c) Display Word Without Legend Instead of using a legend to represent the words, place the word itself at the beginning of each line. The word size is 5

```
# Second version of the 3b plot but without the legend.

ggplot(data=words,mapping =aes(x=Year,y=relative_usage,color=word))+
  geom_line(size=1)+
  xlab("Year")+
  ylab("Relative Word Lsage")+
  theme(legend.position = "none")+
  directlabels::geom_dl(aes(label = word), method = "smart.grid")+



  # Add the function below to make your plot match what is in the homework description
  scale_x_continuous(limits = c(1840, 2020),
                     breaks = seq(1850, 2000, by = 25),
                     minor_breaks = NULL)
```


Relative Word Lsage
75
50
25
1850
1875
1900
1925
1950
1975
2000
Year
idiot
nerd
geek
dork
nimrod
moron