tutorial 8 answers

Leah Brooks

4/8/2021

Homework

1. In my example of DC population over time in section B.1., I present the graph of three steps. Modify your code to make these same three steps.

```
# packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(sf)
```

```
## Linking to GEOS 3.8.0, GDAL 3.0.4, PROJ 6.3.1
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
# load data
counties <- read.csv("h:/pppa_data_viz/2019/tutorial_data/lecture08/counties_1910to2010_20180116.csv")
```

Now just limit the data to DC. You could do this in the `ggplot` call itself. However, in this case when we are only planning to use DC, this gives us a smaller dataset to work with and that speeds processing. This will also make the coding easier, since we won't have to subset in each graph.

Take a look at the data after we subset to DC. Does it have the right number of observations?

```
# get just dc
dct <- counties[which(counties$statefips == 11),]
dim(dct)
```

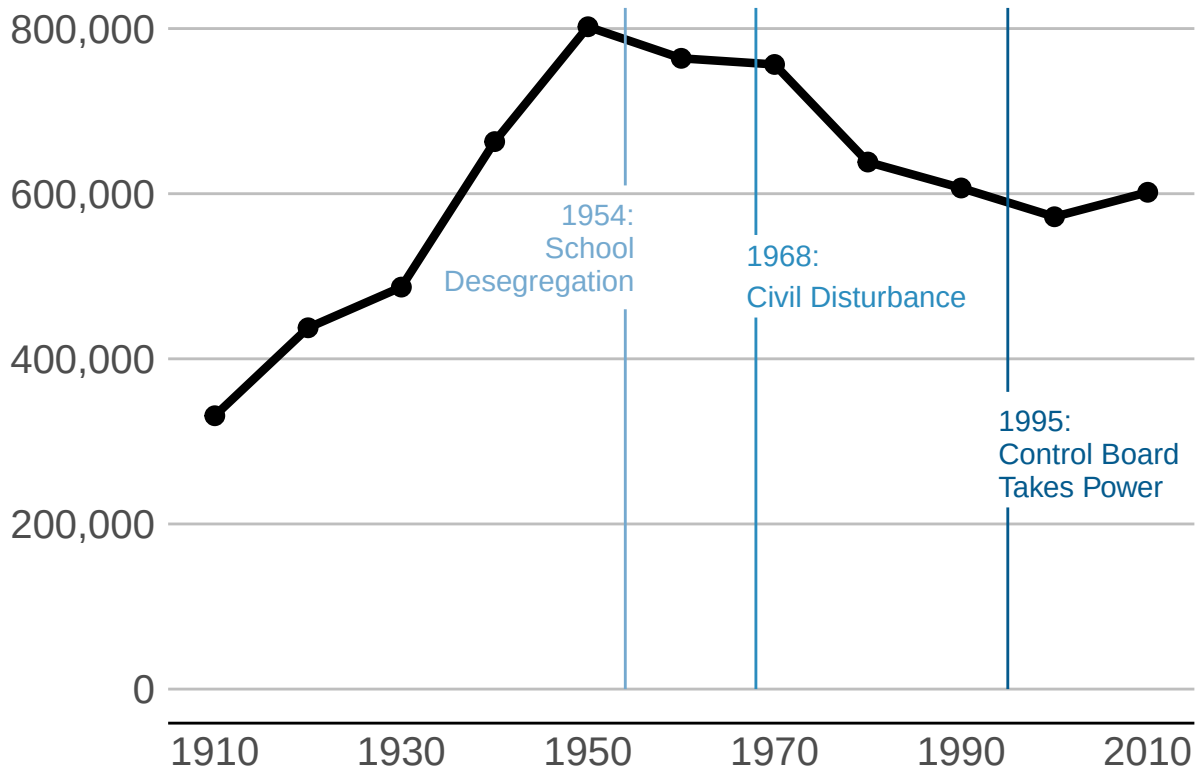
```
## [1] 11 68
```

```
dct[,c("year","statefips","countyfips","cv1")]
```

```
##       year statefips countyfips    cv1
## 285   1910        11          1 331069
## 3244  1920        11          1 437571
## 6314  1930        11          1 486869
## 9418  1940        11          1 663091
## 12520 1950        11          1 802178
## 15626 1960        11          1 763956
## 18764 1970        11          1 756510
## 21899 1980        11          1 638333
## 25039 1990        11          1 606900
## 28182 2000        11          1 572059
## 31326 2010        11          1 601723
```

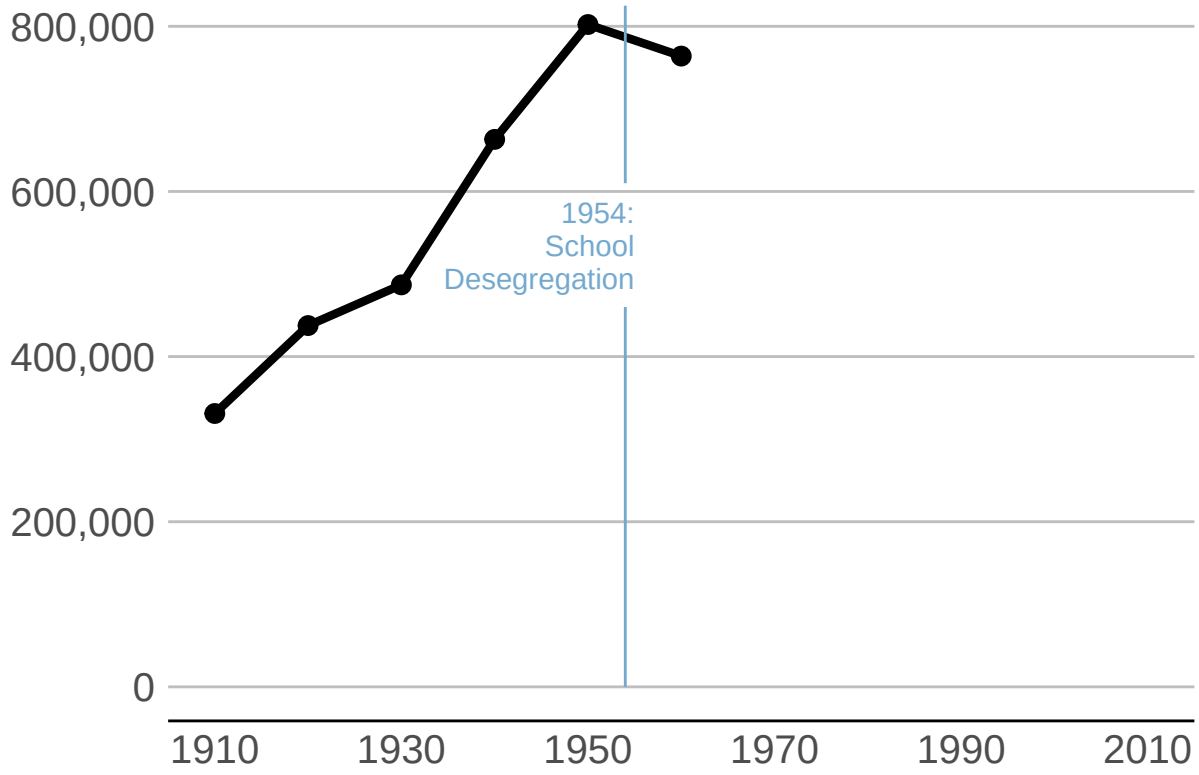
```
# make full graph
on.g.text.size <- 4
done2 <-
  ggplot(dct) +
  geom_line(dct, mapping = aes(x=year, y=cv1), size=1.5) +
  geom_point(dct, mapping = aes(x=year, y=cv1), size=3) +
  scale_y_continuous(labels = comma,  limits = c(0, 825000), breaks = c(seq(0,800000,200000))) +
  scale_x_continuous(limits= c(1910, 2010), breaks = c(seq(1910,2010,20))) +
  labs(x="", y="") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        panel.grid.major.y = element_line(color="gray"),
        legend.position = "none",
        axis.line.x = element_line(color = "black"),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text = element_text(size = 15)) +
  annotate(geom = "segment", x=1995, y=0, xend=1995, yend=220000, color="#045a8d") +
  annotate(geom = "segment", x=1995, y=360000, xend=1995, yend=825000, color="#045a8d") +
  annotate(geom = "segment", x=1968, y=0, xend=1968, yend=450000, color = "#2b8cbe") +
  annotate(geom = "segment", x=1968, y=550000, xend=1968, yend=825000, color = "#2b8cbe") +
  annotate(geom = "segment", x=1954, y=0, xend=1954, yend=460000, color = "#74a9cf") +
  annotate(geom = "segment", x=1954, y=610000, xend=1954, yend=825000, color = "#74a9cf") +
  annotate(geom = "text", x=1955, y=575000, label="1954:", color = "#74a9cf",
           size=on.g.text.size, hjust=1) +
  annotate(geom = "text", x=1955, y=535000, label="School", color = "#74a9cf",
           size=on.g.text.size, hjust=1) +
  annotate(geom = "text", x=1955, y=495000, label="Desegregation", color = "#74a9cf",
           size=on.g.text.size, hjust=1) +
  annotate(geom = "text", x=1967, y=525000, label="1968:", color = "#2b8cbe",
           size=on.g.text.size, hjust=0) +
  annotate(geom = "text", x=1967, y=475000, label="Civil Disturbance", color = "#2b8cbe",
           size=on.g.text.size, hjust=0) +
  annotate(geom = "text", x=1994, y=325000, label="1995:", color="#045a8d",
           size=on.g.text.size, hjust=0) +
  annotate(geom = "text", x=1994, y=285000, label="Control Board", color="#045a8d",
           size=on.g.text.size, hjust=0) +
```

```
  annotate(geom = "text", x=1994, y=245000, label="Takes Power", color="#045a8d",
           size=on.g.text.size, hjust=0)

done2
```




```
# make partial graph
done3 <-
  ggplot() +
  geom_line(dct[which(dct$year < 1970),], mapping = aes(x=year, y=cv1), size=1.5) +
  geom_point(dct[which(dct$year < 1970),], mapping = aes(x=year, y=cv1), size=3) +
  scale_y_continuous(labels = comma,  limits = c(0, 825000), breaks = c(seq(0,800000,200000))) +
  scale_x_continuous(limits= c(1910, 2010), breaks = c(seq(1910,2010,20))) +
  labs(x="", y="") +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        panel.background = element_blank(),
        panel.grid.major.y = element_line(color="gray"),
        legend.position = "none",
        axis.line.x = element_line(color = "black"),
        axis.ticks.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.text = element_text(size = 15)) +
  annotate(geom = "segment", x=1954, y=0, xend=1954, yend=460000, color = "#74a9cf") +
  annotate(geom = "segment", x=1954, y=610000, xend=1954, yend=825000, color = "#74a9cf") +
  annotate(geom = "text", x=1955, y=575000, label="1954:", color = "#74a9cf",
           size=on.g.text.size, hjust=1) +
```

```
  annotate(geom = "text", x=1955, y=535000, label="School", color = "#74a9cf",
           size=on.g.text.size, hjust=1) +
  annotate(geom = "text", x=1955, y=495000, label="Desegregation", color = "#74a9cf",
           size=on.g.text.size, hjust=1)

done3
```



2. Using the bikeshare data,

(a) Re-do one of the by-hour pictures as a minute-by-minute picture showing total ridership
(b) Use one of the y variables we used or an alternative one. Add some annotations to your graph to point out salient features.

```
# load data
cabi.201901 <- read.csv("H:/pppa_data_viz/2019/tutorial_data/lecture08/201902-capitalbikeshare-tripdata/

# check out variables
head(cabi.201901)
```

```
##   Duration          Start.date            End.date Start.station.number
## 1      206 2019-02-01 00:00:20 2019-02-01 00:03:47                31509
## 2      297 2019-02-01 00:04:40 2019-02-01 00:09:38                31203
## 3      165 2019-02-01 00:06:34 2019-02-01 00:09:20                31303
## 4      176 2019-02-01 00:06:49 2019-02-01 00:09:45                31400
## 5      105 2019-02-01 00:10:41 2019-02-01 00:12:27                31270
## 6      757 2019-02-01 00:12:37 2019-02-01 00:25:14                31503
##                               Start.station End.station.number
## 1                   New Jersey Ave & R St NW             31636
```

```
## 2                   14th & Rhode Island Ave NW              31519
## 3 Tenleytown / Wisconsin Ave & Albemarle St NW              31308
## 4               Georgia & New Hampshire Ave NW              31401
## 5                                 8th & D St NW             31256
## 6                          Florida Ave & R St NW            31126
##                           End.station Bike.number Member.type
## 1 New Jersey Ave & N St NW/Dunbar HS      W21713      Member
## 2                       1st & O St NW      E00013      Member
## 3                  39th & Veazey St NW      W21703      Member
## 4              14th St & Spring Rd NW       W21699      Member
## 5                     10th & E St NW        W21710      Member
## 6                  11th & Girard St NW      W22157      Member
```

```
# preapre time variables
cabi.201901$time.start <- as.POSIXct(strptime(x = cabi.201901$Start.date,
                                              format = "%Y-%m-%d %H:%M:%S"))
cabi.201901$time.stop  <- as.POSIXct(strptime(x = cabi.201901$End.date,
                                              format = "%Y-%m-%d %H:%M:%S"))

# my duration calculation
cabi.201901$my.duration <- cabi.201901$time.stop - cabi.201901$time.start
cabi.201901$Duration.minutes <- cabi.201901$Duration / 60
summary(cabi.201901$Duration.minutes)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    1.000    5.817    9.617   14.931   15.950 1435.000
```

```
# get the minute out of the date variable
cabi.201901$start.minute <- as.numeric(format(cabi.201901$time.start, "%M"))
summary(cabi.201901$start.minute)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00   15.00   30.00   29.59   45.00   59.00
```

```
table(cabi.201901$start.minute)
```

```
##
##    0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2427 2538 2537 2590 2594 2751 2700 2675 2658 2634 2601 2721 2655 2583 2637 2597
##   16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 2747 2776 2651 2619 2624 2586 2716 2654 2595 2589 2510 2456 2435 2471 2541 2565
##   32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47
## 2534 2661 2623 2673 2702 2613 2724 2745 2776 2745 2872 2701 2759 2685 2689 2732
##   48   49   50   51   52   53   54   55   56   57   58   59
## 2639 2632 2746 2742 2649 2702 2606 2632 2603 2478 2501 2533
```

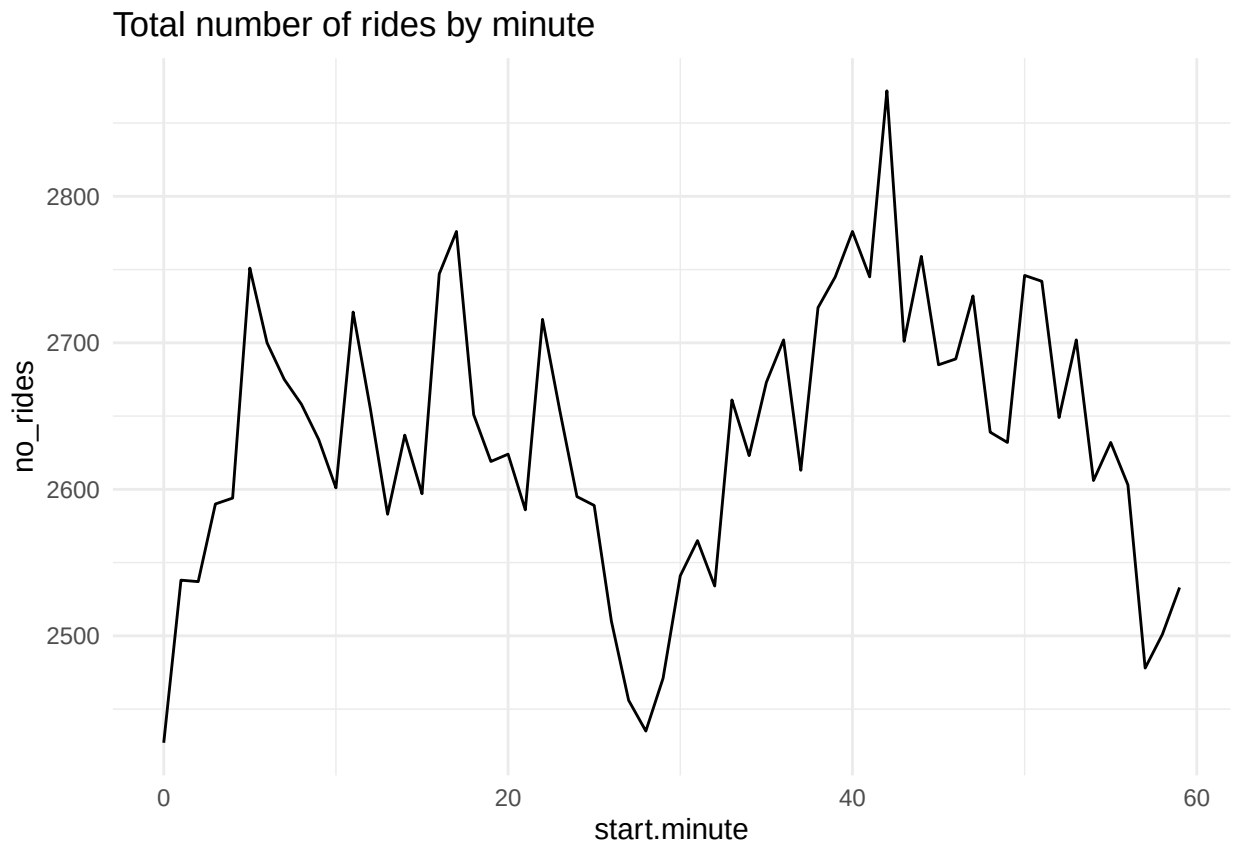
```
# make an indicator for a member
cabi.201901$member <- ifelse(cabi.201901$Member.type == "Member", 1, 0)

# summarize to minute data
cabi.201901 <- group_by(cabi.201901, start.minute)
cabisum <- summarize(.data = cabi.201901, no_rides = n(),
                                          mean_dur = mean(Duration),
                                          member_rides = sum(member))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

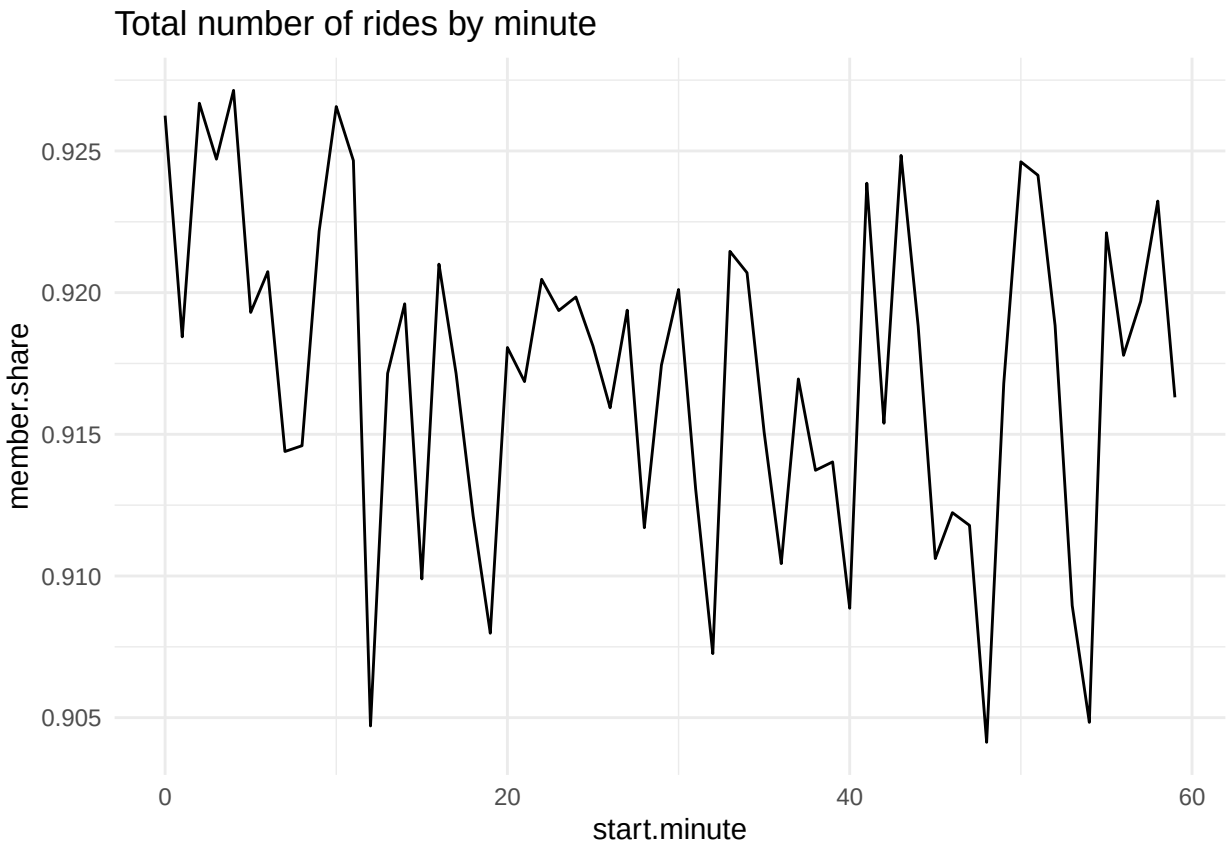
```r
dim(cabisum)
```

```
## [1] 60  4
```

```r
# find member share of rides
cabisum$member.share <- cabisum$member_rides / cabisum$no_rides

# number of rides by minute
c3 <- ggplot() +
  geom_line(data = cabisum, mapping = aes(x = start.minute, y = no_rides)) +
  labs(title = "Total number of rides by minute") +
  theme_minimal()
c3
```




```r
# member share of rides by minute
c4 <- ggplot() +
  geom_line(data = cabisum, mapping = aes(x = start.minute, y = member.share)) +
  labs(title = "Total number of rides by minute") +
  theme_minimal()
c4
```



3. More stacked areas

Now you try to load your own budget data!

Use Table 1.3 (his01z3.xls), from which we want the year and columns E, F, G and columns I, J and K. Create a new excel document with just this information, and make one row at top with names that you'll understand. Keep just through 2017, and make sure that you don't have any junk at the bottom of the table. Save this file as csv (file, save as, choose "csv" option for file type).

Load it into R and make a stacked area graph of receipts, outlays and deficits over time.

Having done this myself, here are a few suggestions

- make long data, as we did above
- make year numeric, as we did for the social insurance revenue above
- get rid of commas in the data. My command to do this, for one variable, is

```
hist01z3$b1 <- as.numeric(gsub(",", "", hist01z3$cd.receipts, fixed = TRUE))
```

```
library(tidyverse)

### receipts/surplus/deficits constant dollars ####
hist01z3 <- read.csv("H:/pppa_data_viz/2018/tutorials/lecture05/omb_data/hist01z3.csv")

names(hist01z3)
```

```
## [1] "year"        "cd.receipts" "cd.outlays"  "cd.surplus"  "pg.receipts"
## [6] "pg.outlays"  "pg.surplus"
```

```
### need to make this long
table(hist01z3$year)
```

```
##
## 1940 1941 1942 1943 1944 1945 1946 1947 1948 1949 1950 1951 1952 1953 1954 1955
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 1956 1957 1958 1959 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1971
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 1988 1989 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
## 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017   TQ
##    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
```

```
# clean up variables for reshape
hist01z3$nyear <- as.numeric(levels(hist01z3$year))[hist01z3$year]
summary(hist01z3$nyear)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      NA      NA      NA     NaN      NA      NA      79
```

```
# rename and make numeric for reshape
# warning: you also need to get rid of commas in the numbers
# see http://rfunction.com/archives/2354
hist01z3$b1 <- as.numeric(gsub(",", "", hist01z3$cd.receipts, fixed = TRUE))
hist01z3$b2 <- as.numeric(gsub(",", "", hist01z3$cd.outlays, fixed = TRUE))
hist01z3$b3 <- as.numeric(gsub(",", "", hist01z3$cd.surplus, fixed = TRUE))
# make a negative outlays for a more interesting chart
hist01z3$b4 <- hist01z3$b2 * -1

# just keep the variabls we make long
hist2 <- hist01z3[,c("year","b1","b2","b3","b4")]

# reshape to long
b.long <- pivot_longer(data = hist2,
                       cols = c("b1", "b2","b3","b4"),
                       names_to = "btype",
                       values_to = "nyear")
b.long[1:15,]
```

```
## # A tibble: 15 x 3
##    year  btype  nyear
##    <chr> <chr>  <dbl>
##  1 1940  b1      94
##  2 1940  b2     136.
##  3 1940  b3     -41.9
##  4 1940  b4    -136.
##  5 1941  b1     113.
##  6 1941  b2     178.
##  7 1941  b3     -64.3
##  8 1941  b4    -178.
##  9 1942  b1     168.
## 10 1942  b2     403.
## 11 1942  b3    -235.
```

```
## 12 1942  b4    -403.
## 13 1943  b1     249
## 14 1943  b2     815.
## 15 1943  b3    -566.
```

```
# give names for types
# make a type of receipts variable
b.long$bname <- ifelse(b.long$btype == "b1","receipts",
                       ifelse(b.long$btype == "b2", "outlays",
                              ifelse(b.long$btype == "b3", "surplus",
                                     ifelse(b.long$btype == "b4","outlays",""))))

sub.long <- b.long[which(b.long$btype %in% c("year","nyear","b1","b4","b3")),]
table(sub.long$bname)
```

```
##
##  outlays receipts  surplus
##       79       79       79
```

```
sub.long$bname.fac <- as.factor(sub.long$bname)
levels(sub.long$bname.fac)
```

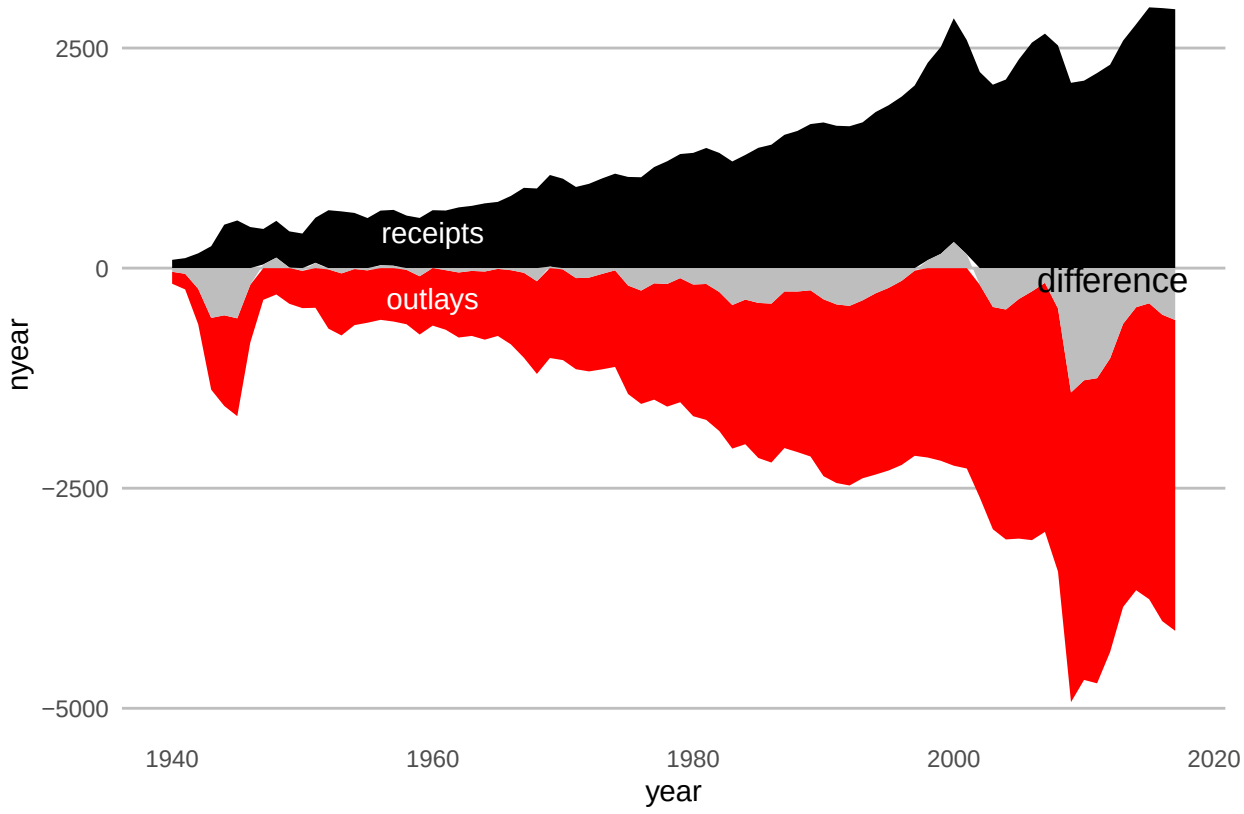
```
## [1] "outlays"  "receipts" "surplus"
```

```
# make year numeric
sub.long$year <- as.numeric(sub.long$year)
```

```
## Warning: NAs introduced by coercion
```

```
### up and down chart of receipts/surplus/deficits ###
#### stacked chart of total receipts by type ###
## without factor() this doesnt work


hw5q2 <-
  ggplot() +
  geom_area(data = sub.long,
            mapping = aes(x=year, y=nyear,
                          group = bname.fac, fill=bname.fac)) +
  scale_fill_manual(values = c("red", "black","grey")) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), panel.grid.major.y = element_line(color="gray"),
        axis.ticks.x = element_blank(), axis.ticks.y = element_blank(),
        legend.position = "none") +
  annotate("text", x=1960, y=400, label="receipts", color = "white") +
  annotate("text", x=1960, y=-350, label="outlays", color = "white") +
  annotate("text", x=2012.2, y=-130, label="difference", color = "black", size = 4.5)

hw5q2
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```




```
fn <- paste0("H:/pppa_data_viz/2021/tutorials/tutorial_08/lecture08_line_charts_answers_v02_files/final
             "20210408.jpg")

ggsave(filename = fn,
       plot = hw5q2,
       device = c("jpg"),
       height = 6,
       width = 8,
       units = c("in"))
```

```
## Warning: Removed 3 rows containing missing values (position_stack).
```