p8105_hw3_pl2811

Pei Hsin Lin

10/18/2021

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.2     v forcats 0.5.1

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

```
library(dplyr)
library(ggridges)
library(ggplot2)
library(forcats)
library(p8105.datasets)
library(httr)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

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

```
library(viridis)
```

```
## Loading required package: viridisLite
```

```
library(patchwork)
library(knitr)
library(png)
```

###How many aisles are there, and which aisles are the most items ordered from?

```
data("instacart")
skimr::skim(aisle_count)
```

```
aisle_count<-instacart %>%
    group_by(aisle) %>%
    count()
```

#There are 134 aisles

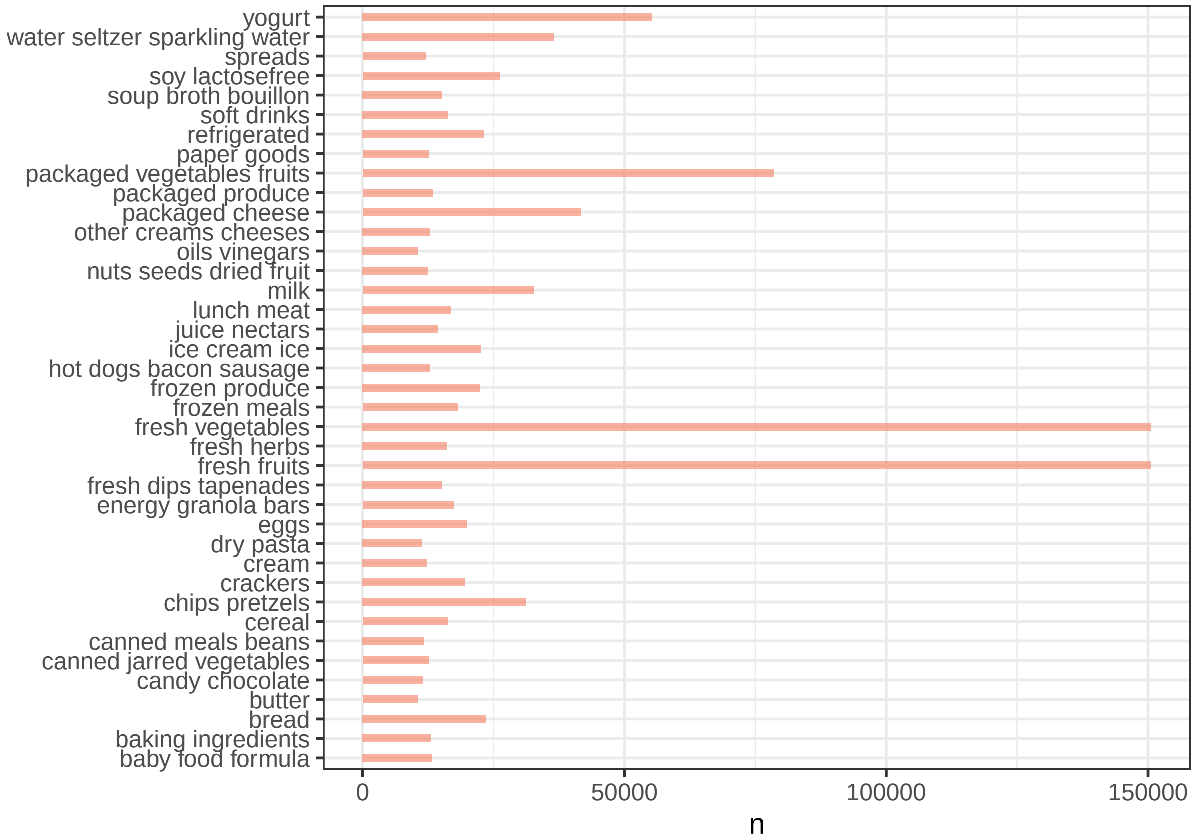
```
 aisle_order= order(aisle_count$n, decreasing=T)
 aisle_sorted = aisle_count[aisle_order,]
 head(aisle_sorted, 1)
```

```
## # A tibble: 1 x 2
## # Groups:   aisle [1]
##   aisle                  n
##   <chr>              <int>
## 1 fresh vegetables 150609
```

most items ordered from fresh vegetables

Make a plot that shows the number of items ordered in each aisle, limiting this to aisles with more than 10000 items ordered. Arrange aisles sensibly, and organize your plot so others can read it.

```
instacart_order<-filter(aisle_sorted, n > 10000)
instacart_order %>%
mutate(name = fct_reorder(aisle, desc(n))) %>%
ggplot( aes(x=aisle, y=n)) +
geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4) +
coord_flip() +
xlab("") +
theme_bw()
```



Make a table showing the three most popular items in each of the aisles "baking ingredients", "dog food care", and "packaged vegetables fruits". Include the number of times each item is ordered in your table.

```
popular_items<-select(instacart, aisle, product_name)

popular_items<-filter(popular_items, aisle %in% c("dog food care", "baking ingredients",
                                                  "packaged vegetables fruits" ))

popular_items_tables<-popular_items %>%
    group_by(product_name, aisle) %>%
    count()

aisle_popular<-popular_items_tables %>%
    group_by(aisle) %>%
    top_n(1, n)

aisle_popular
```

```
## # A tibble: 3 x 3
## # Groups:   aisle [3]
##   product_name                                 aisle                          n
##   <chr>                                        <chr>                      <int>
## 1 Light Brown Sugar                            baking ingredients           499
```

```
## 2 Organic Baby Spinach                         packaged vegetables fruits  9784
## 3 Snack Sticks Chicken & Rice Recipe Dog Treats dog food care                30
```

Make a table showing the mean hour of the day at which Pink Lady Apples and Coffee Ice Cream are ordered on each day of the week;

format this table for human readers (i.e. produce a 2 x 7 table)

```
week<-filter(instacart, product_name %in% c("Pink Lady Apples", "Coffee Ice Cream"))
week<-select(week, order_dow, order_hour_of_day, product_name )
week<- week %>%
  group_by(order_dow, product_name ) %>%
  summarise_at(vars(order_hour_of_day), list(mean= mean))

week<-pivot_wider(
 week,
  names_from = "product_name",
  values_from = "mean")

 week= subset(week, select = -c( order_dow ))
week
```

```
## # A tibble: 7 x 2
##   `Coffee Ice Cream` `Pink Lady Apples`
##                <dbl>              <dbl>
## 1               13.8               13.4
## 2               14.3               11.4
## 3               15.4               11.7
## 4               15.3               14.2
## 5               15.2               11.6
## 6               12.3               12.8
## 7               13.8               11.9
```

```
library(p8105.datasets)
data("brfss_smart2010")
brfss_smart2010=brfss_smart2010%>%
  janitor::clean_names()

overall_health<-filter(brfss_smart2010,topic %in% c("Overall Health"))
overall_health<-filter(brfss_smart2010, response %in% c("Poor","Fair","Good","Very good", "Excellent"))
topic<-overall_health %>%
    group_by(response) %>%
    count()


target <- c("Poor","Fair","Good","Very good", "Excellent")
overall_health<-overall_health[order(factor(overall_health$response, levels = target)),]
```

#In 2002, which states were observed at 7 or more locations? What about in 2010?

```
overall_health_2002<-filter(overall_health, year %in% c("2002"))
location<- overall_health_2002 %>%
  group_by(locationabbr)%>%
    count()

location02_over_7<-filter(location, n>=7)
view(location02_over_7)
knitr::kable(location02_over_7, "pipe")
```

locationabbr	n
AZ	10
CO	20
CT	35
DE	15
FL	35
GA	15
HI	20
ID	10
IL	15
IN	10
KS	15
LA	15
MA	40
MD	30
ME	10
MI	20
MN	20
MO	10
NC	35
NE	15
NH	25
NJ	40
NV	10
NY	25
OH	20
OK	15
OR	15
PA	50
RI	20
SC	15
SD	10
TN	10
TX	10
UT	25
VT	15
WA	20

```
overall_health_2010<-filter(overall_health, year %in% c("2010"))
location<- overall_health_2010 %>%
  group_by(locationabbr)%>%
    count()
```

```r
location10_over_7<-filter(location, n>=7)
knitr::kable(location10_over_7, "pipe")
```

locationabbr	n
AL	15
AR	15
AZ	15
CA	60
CO	35
CT	25
DE	15
FL	205
GA	20
HI	20
IA	10
ID	30
IL	10
IN	15
KS	20
LA	25
MA	45
MD	60
ME	30
MI	20
MN	25
MO	15
MS	10
MT	15
NC	60
ND	15
NE	50
NH	25
NJ	95
NM	30
NV	10
NY	45
OH	40
OK	15
OR	20
PA	35
RI	25
SC	35
SD	10
TN	25
TX	80
UT	30
VT	30
WA	50
WY	10

Table location02__over__7 shows the 36 states were observed at 7 or more locations in 2002.

Table location10__over__7 shows the 36 states were observed at 7 or more locations in 2010.

###Construct a dataset that is limited to Excellent responses, and contains, year, state, and a variable that averages the data_value across locations within a state.

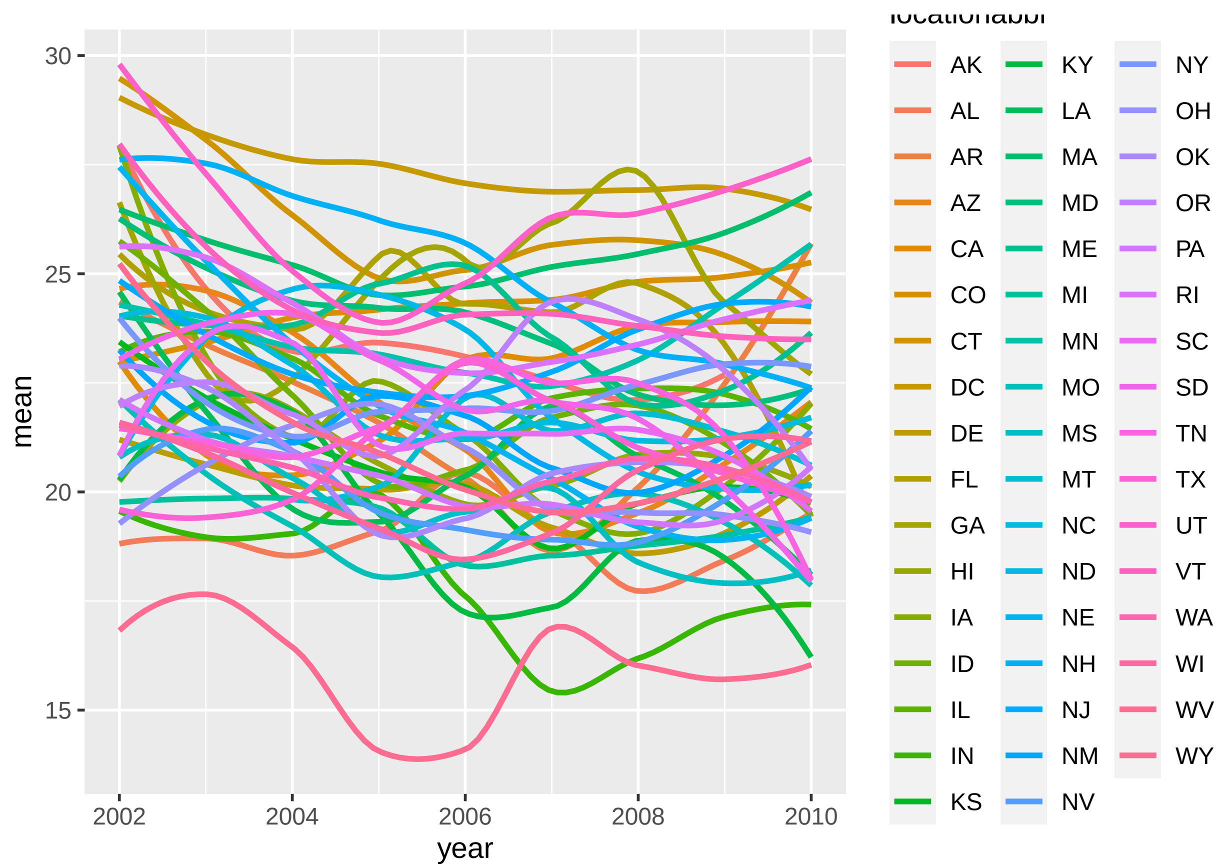
```
brfss_s<-select(brfss_smart2010, year, locationabbr,locationdesc,response, data_value)

brfss_s<-filter(brfss_s, response%in% c("Excellent"))
brfss_s<- na.omit(brfss_s)

brfss_ss<- brfss_s %>%
group_by(locationabbr, year) %>%
summarise_at(vars(data_value), list(mean= mean))
brfss_ss
```

```
## # A tibble: 443 x 3
## # Groups:   locationabbr [51]
##    locationabbr  year  mean
##    <chr>        <int> <dbl>
##  1 AK            2002  27.9
##  2 AK            2003  24.8
##  3 AK            2004  23.0
##  4 AK            2005  23.8
##  5 AK            2007  23.5
##  6 AK            2008  20.6
##  7 AK            2009  23.2
##  8 AL            2002  18.5
##  9 AL            2003  19.5
## 10 AL            2004  20
## # ... with 433 more rows
```

###Make a “spaghetti” plot of this average value over time within a state (that is, make a plot showing a line for each state across years – the geom_line geometry and group aesthetic will help).

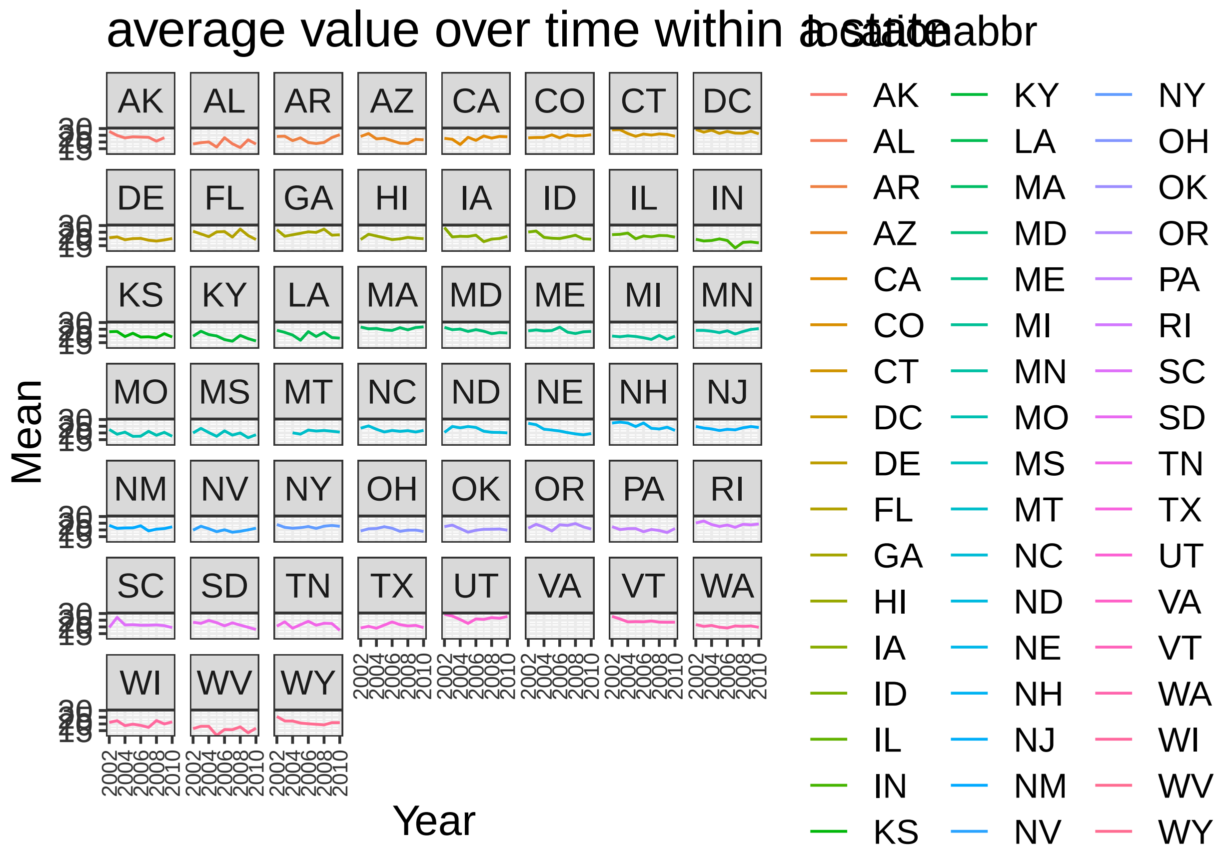
```
ggplot(brfss_ss, aes(x = year, y = mean, color =locationabbr)) +
  geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
ggplot(data =brfss_ss, aes(x = year, y = mean, color = locationabbr)) +
 geom_line() +
facet_wrap(~ locationabbr)+
      labs(title = 'average value over time within a state',
        x = 'Year',
        y = 'Mean') +
    theme_bw() +
    theme(axis.text.x = element_text(colour="grey20", size=8, angle=90, hjust=.5, vjust=.5),
                          axis.text.y = element_text(colour="grey20", size=12),
          text=element_text(size=16))
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

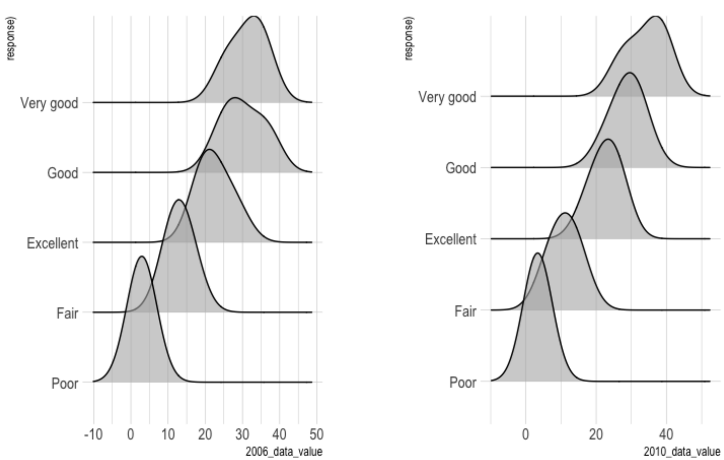


#since this spaghetti plot is too complicate to read,we separate each states to made it more easy to read.

#Make a two-panel plot showing, for the years 2006, and 2010, distribution of data_value for responses ("Poor" to "Excellent") among locations in NY State.

```
ny06<-filter(brfss, locationabbr%in% c("NY"),year==2006)
ny06<-filter(ny06, response %in% c("Poor","Fair","Good","Very good", "Excellent"))

a<-ny06 %>%
  mutate(response= fct_reorder(response, data_value)) %>%
  ggplot( aes(y=response, x=data_value)) +
    geom_density_ridges(alpha=0.6, bandwidth=4) +
    scale_fill_viridis(discrete=TRUE) +
    scale_color_viridis(discrete=TRUE) +
    theme_ipsum() +
    theme(
      legend.position="none",
      panel.spacing = unit(0.1, "lines"),
      strip.text.x = element_text(size = 8)
    ) +
  ylab("response)")+
  xlab("2006_data_value")



ny10<-filter(brfss, locationabbr%in% c("NY"),year==2010 )
```

```
ny10<-filter(ny10, response %in% c("Poor","Fair","Good","Very good", "Excellent"))


b<-ny10 %>%
  mutate(response= fct_reorder(response, data_value)) %>%
  ggplot( aes(y=response, x=data_value)) +
    geom_density_ridges(alpha=0.6, bandwidth=4) +
    scale_fill_viridis(discrete=TRUE) +
    scale_color_viridis(discrete=TRUE) +
    theme_ipsum() +
    theme(
      legend.position="none",
      panel.spacing = unit(0.1, "lines"),
      strip.text.x = element_text(size = 8)
    ) +
  ylab("response)")+
  xlab("2010_data_value")

a+b
#There are some error invovle theme_ipsum and R markdown so I use eval=FALSE. However the code run well
```




```
accel_data<- read.csv(file ="/Users/lin/Desktop/accel_data.csv")
```

#Load, tidy, and otherwise wrangle the data. Your final dataset should include all originally observed variables and values; have useful variable names; include a weekday vs weekend variable; and encode data with reasonable variable classes. Describe the resulting dataset (e.g. what variables exist, how many observations, etc).

```
accel_data<-mutate(accel_data, week_d= ifelse(day== "Sunday", "weekend",
                                              (ifelse(day== "Sunday",
                                                      "weekend", "weekday"))))
```

There are 35 observations, represents 35 days of trials. There are 1444 variables, other than the week_d variable, we also have week, day_id, day, and activity.1 to activity.1440(as 1440 minutes/ 24 hours of one day).

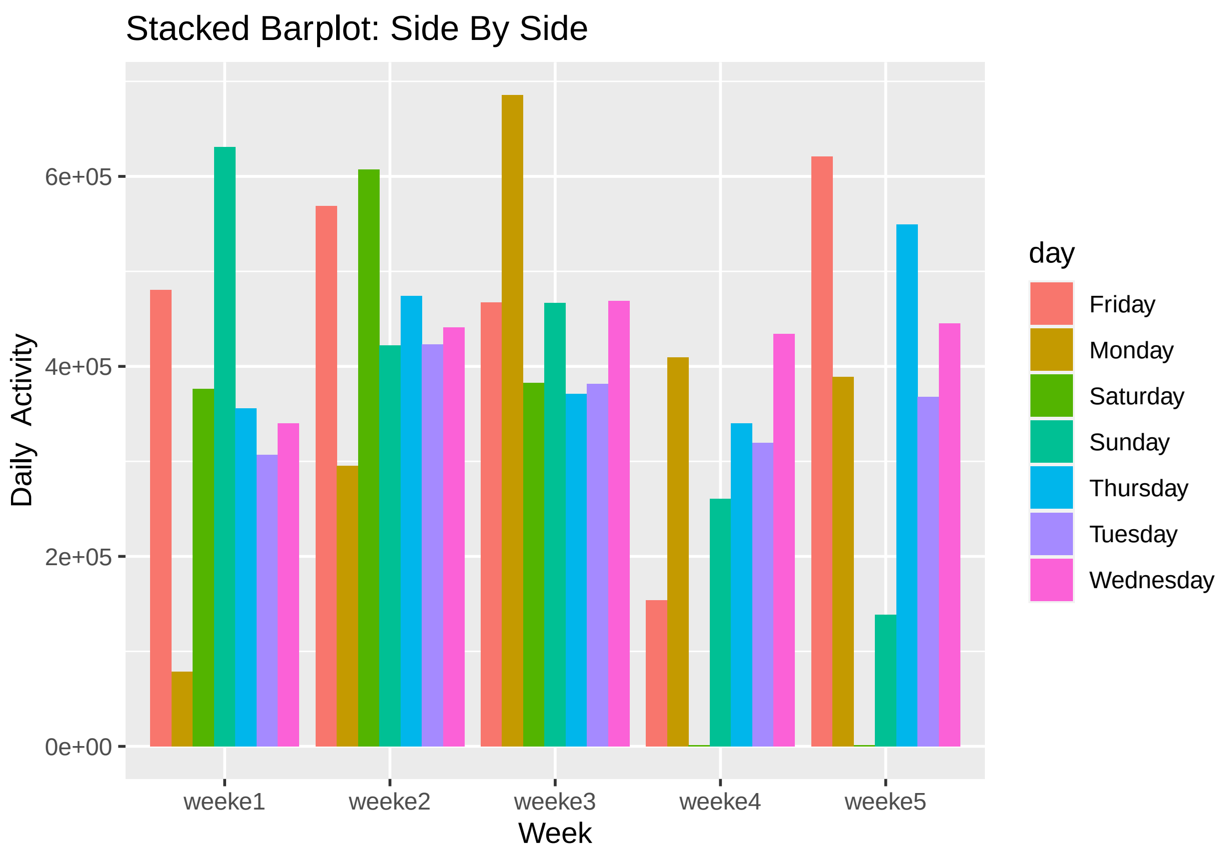
###Traditional analyses of accelerometer data focus on the total activity over the day. Using your tidied dataset, aggregate accross minutes to create a total activity variable for each day, and create a table showing these totals. Are any trends apparent?

```
accel_data<-accel_data%>%
    mutate(total = select(.,activity.1:activity.1440) %>%
               rowSums(na.rm = TRUE))

total_act<-select(accel_data,day_id, total)
#since Table is hard to observed any trends , we make plot to see is any trends apparent?
p <- ggplot(data = accel_data, aes(x = day_id, y = total)) +
     geom_line(color = "#00AFBB", size = 1)
accel_data%>%
  ggplot( aes(x=day_id)) +
    geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)+
     theme_ipsum()

total_act
```

#didn't observe particular trends apparent

###Accelerometer data allows the inspection activity over the course of the day. Make a single-panel plot that shows the 24-hour activity time courses for each day and use color to indicate day of the week. Describe in words any patterns or conclusions you can make based on this graph.

```
accel_data<-accel_data%>%
    mutate(total = select(.,activity.1:activity.1440) %>%
               rowSums(na.rm = TRUE))

total_act<-select(accel_data,day_id, total)

accel_data<-mutate(accel_data, weeknum = ifelse(day_id < 8, "weeke1",
                                            ifelse(day_id < 15, "weeke2",
                                            ifelse(day_id < 22 , "weeke3",
                                            ifelse(day_id < 29 , "weeke4","weeke5")))))

accel_data %>%
  ggplot(aes( weeknum,total, fill= day))+
  geom_col(position="dodge") +
  labs(title="Stacked Barplot: Side By Side",
        x="Week", y= "Daily  Activity")
```


Stacked Barplot: Side By Side
Daily Activity
6e+05
4e+05
2e+05
0e+00
weeke1
weeke2
weeke3
weeke4
weeke5
Week
day
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday