PHW251 Problem Set 5

Teaching Team

At this point in the course we have introduced a fair amount of code, which can be a lot to hold in our memory at once! Thankfully we we have search engines and these helpful cheatsheets. You may find the Base R and Data Transformation Cheatsheet helpful.

Part 1

Question 1

Use the readxl library and load two data sets from the "two_data_sheets" file. There's a parameter that you can specify which sheet to load. In this case, we have data about rat reaction time in sheet 1 and home visits in sheet 2.

```
# your code here
library(readxl)
df_rats <- read_excel("data/two_data_sheets.xlsx", 1)
df_home <- read_excel("data/two_data_sheets.xlsx", 2)
```

Question 2

For the rats data, pivot the data frame from wide to long format. We want the 1, 2, 3 columns, which represent the amount of cheese placed in a maze, to transform into a column called "cheese". The values in the cheese column will be the time, which represents the amount of time the rat took to complete the maze. Please use the `head()` function to print the first few rows of your data frame.

```
# your code here
df_rats$subject <- factor(df_rats$subject)
df_rats_long <- df_rats %>%
  pivot_longer(c(`1`, `2`, `3`), names_to = "cheese", values_to = "time")

head(df_rats_long)
```

```
## # A tibble: 6 x 3
##   subject cheese  time
##   <fct>   <chr>  <dbl>
## 1 rat_101 1      14.4
## 2 rat_101 2       9.01
## 3 rat_101 3       8.20
## 4 rat_102 1      11.7
## 5 rat_102 2       8.59
## 6 rat_102 3       8.49
```

```
df_rats_wide <- df_rats_long %>%
  pivot_wider(names_from = "cheese", values_from = "time")
```

Question 3

Use `summarize()` to compute the mean and standard deviation of the maze time depending on the amount of cheese in the maze.

```
# your code here
df_rats_long %>%
  # organize by amount of cheese
  group_by(cheese) %>%
  # summarize
  summarize(mean = mean(time), # mean function
            sd = sd(time))     # standard deviation function
```

```
## # A tibble: 3 x 3
##   cheese  mean    sd
##   <chr>  <dbl> <dbl>
## 1 1      12.8  1.43
## 2 2       9.88 0.904
## 3 3       8.51 0.279
```

Question 3

The home visits data is a record of how and where some interviews were conducted. Pivot the home visits data frame from long to wide. We want the names from the action column to become unique columns and the values to represent the counts. Please print your whole resulting data frame.

```
# your code here
df_home_wide <- df_home %>%
  pivot_wider(names_from = action, values_from = count)

df_home_wide
```

```
## # A tibble: 9 x 5
##   location       year interview `home visit` questionnaire
##   <chr>         <dbl>     <dbl>        <dbl>         <dbl>
## 1 Washington DC  2015       103           76           200
## 2 Washington DC  2016        71           43           168
## 3 Washington DC  2017        45           60            90
## 4 St Louis       2015        90           86           210
## 5 St Louis       2016        95           82           175
## 6 St Louis       2017        78           71           106
## 7 Tucson         2015       130           98           303
## 8 Tucson         2016       120           88           280
## 9 Tucson         2017        78           65           230
```

Part 2

For this part we will use data from New York City that tested children under 6 years old for elevated blood lead levels (BLL). [You can read more about the data on their website]).

About the data:

All NYC children are required to be tested for lead poisoning at around age 1 and age 2, and to be screened for risk of lead poisoning, and tested if at risk, up until age 6. These data are an indicator of children younger that 6 years of age tested in NYC in a given year with blood lead levels (BLL) of 5 mcg/dL or greater. In 2012, CDC established that a blood lead level of 5 mcg/dL is the reference level for exposure to lead in children. This level is used to identify children who have blood lead levels higher than most children's levels. The reference level is determined by measuring the NHANES blood lead distribution in US children ages 1 to 5 years, and is reviewed every 4 years.

Question 4

Recreate the below table with the "kable" pacakge.

```
knitr::include_graphics('data/question_1_table.png')
```

BLL Rates per 1,000 tested in New York City, 2015-2016

Borough	Year	BLL >5 µg/dL	BLL >10 µg/dL	BLL >15 µg/dL
Bronx	2015	15.7	2.5	1.0
Bronx	2016	15.0	2.8	1.2
Brooklyn	2015	22.6	3.9	1.3
Brooklyn	2016	22.3	3.6	1.2
Manhattan	2015	10.6	1.6	0.5
Manhattan	2016	8.1	1.3	0.6
Queens	2015	15.4	2.7	1.0
Queens	2016	14.3	2.3	0.9
Staten Island	2015	12.0	2.0	0.7
Staten Island	2016	14.8	2.7	0.8

You will need to calculate the BLL per 1,000, filter for years 2015-2016, and rename the boroughs based on the following coding scheme:

- 1: Bronx
- 2: Brooklyn
- 3: Manhattan
- 4: Queens
- 5: Staten Island

First, filter your dataframe for the years 2015-2016 and rename the boroughs. If you make your borough names a factor, it will make your life easier when we create tables and graphs.

```
# your code here
bll_nyc2 <- bll_nyc %>%
  filter(time_period %in% c("2015", "2016")) %>%
  mutate(borough_id = factor(borough_id,
                             levels = c(1:5),
                             labels = c("Bronx","Brooklyn", "Manhattan",
                                        "Queens", "Staten Island"),
                             ordered=TRUE))

head(bll_nyc2)
```

```
## # A tibble: 6 x 6
##   borough_id time_period bll_5 bll_10 bll_15 total_tested
##   <ord>            <dbl> <dbl>  <dbl>  <dbl>        <dbl>
## 1 Bronx             2015   971    155     61        61700
## 2 Bronx             2016   884    162     71        59000
## 3 Brooklyn          2015  2458    423    142       108800
## 4 Brooklyn          2016  2314    376    122       103800
## 5 Manhattan         2015   399     59     19        37500
## 6 Manhattan         2016   289     46     22        35800
```

Second, group and summarize the data to calculate the total *number* of children in each borough in each year that were tested and the number with blood lead levels that were greater than 5 mcg/dL, 10 5 mcg/dL, and 15 5 mcg/dL.

```
# your code here
bll_nyc3 <- bll_nyc2 %>%
  group_by(borough_id, time_period) %>%
  summarize(total_tested = sum(total_tested),
            bll_5 = sum(bll_5),
            bll_10 = sum(bll_10),
            bll_15 = sum(bll_15))
```

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

```
bll_nyc3
```

```
## # A tibble: 10 x 6
## # Groups:   borough_id [5]
##    borough_id    time_period total_tested bll_5 bll_10 bll_15
##    <ord>               <dbl>        <dbl> <dbl>  <dbl>  <dbl>
##  1 Bronx                2015       123100  1937    310    122
##  2 Bronx                2016       117800  1763    324    142
##  3 Brooklyn             2015       217400  4911    846    284
##  4 Brooklyn             2016       207500  4627    752    244
##  5 Manhattan            2015        74000   787    118     38
##  6 Manhattan            2016        70400   567     92     44
##  7 Queens               2015       178900  2750    488    174
```

```
##  8 Queens               2016       174600  2490    406    150
##  9 Staten Island        2015        27400   328     54     18
## 10 Staten Island        2016        25900   384     70     20
```

Third, calculate the rate at which each blood lead level occurred in each year in each borough (BLL per 1,000).

```
# your code here
bll_nyc4 <- bll_nyc3 %>% mutate(bll_5_per_1k = round(bll_5/total_tested * 1000, 1),
                                bll_10_per_1k = round(bll_10/total_tested * 1000, 1),
                                bll_15_per_1k = round(bll_15/total_tested * 1000, 1))

bll_nyc4
```

```
## # A tibble: 10 x 9
## # Groups:   borough_id [5]
##    borough_id    time_period total_tested bll_5 bll_10 bll_15 bll_5_per_1k
##    <ord>               <dbl>        <dbl> <dbl>  <dbl>  <dbl>        <dbl>
##  1 Bronx                2015       123100  1937    310    122         15.7
##  2 Bronx                2016       117800  1763    324    142         15
##  3 Brooklyn             2015       217400  4911    846    284         22.6
##  4 Brooklyn             2016       207500  4627    752    244         22.3
##  5 Manhattan            2015        74000   787    118     38         10.6
##  6 Manhattan            2016        70400   567     92     44          8.1
##  7 Queens               2015       178900  2750    488    174         15.4
##  8 Queens               2016       174600  2490    406    150         14.3
##  9 Staten Island        2015        27400   328     54     18         12
## 10 Staten Island        2016        25900   384     70     20         14.8
## # i 2 more variables: bll_10_per_1k <dbl>, bll_15_per_1k <dbl>
```

Now we have calculated all the numbers we need to recreate the table shown at the beginning of this question. Use `kable()` to produce your table.

```
# your code here

# select columns and change the year to character so it doesn't get a big.mark
bll_nyc5 <- bll_nyc4 %>%
  select(borough_id, time_period, bll_5_per_1k, bll_10_per_1k, bll_15_per_1k) %>%
  mutate(time_period = as.character(time_period))

kable(bll_nyc5,
      booktabs=T,
      col.names=c("Borough", "Year", "BLL >5 µg/dL", "BLL >10 µg/dL", "BLL >15 µg/dL"),
      align='lcccc',
      caption="BLL Rates per 1,000 tested in New York City, 2015-2016",
      format.args=list(big.mark=","))
```

Table 1: BLL Rates per 1,000 tested in New York City, 2015-2016

Borough	Year	BLL >5 µg/dL	BLL >10 µg/dL	BLL >15 µg/dL
Bronx	2015	15.7	2.5	1.0

Borough	Year	BLL >5 µg/dL	BLL >10 µg/dL	BLL >15 µg/dL
Bronx	2016	15.0	2.8	1.2
Brooklyn	2015	22.6	3.9	1.3
Brooklyn	2016	22.3	3.6	1.2
Manhattan	2015	10.6	1.6	0.5
Manhattan	2016	8.1	1.3	0.6
Queens	2015	15.4	2.7	1.0
Queens	2016	14.3	2.3	0.9
Staten Island	2015	12.0	2.0	0.7
Staten Island	2016	14.8	2.7	0.8

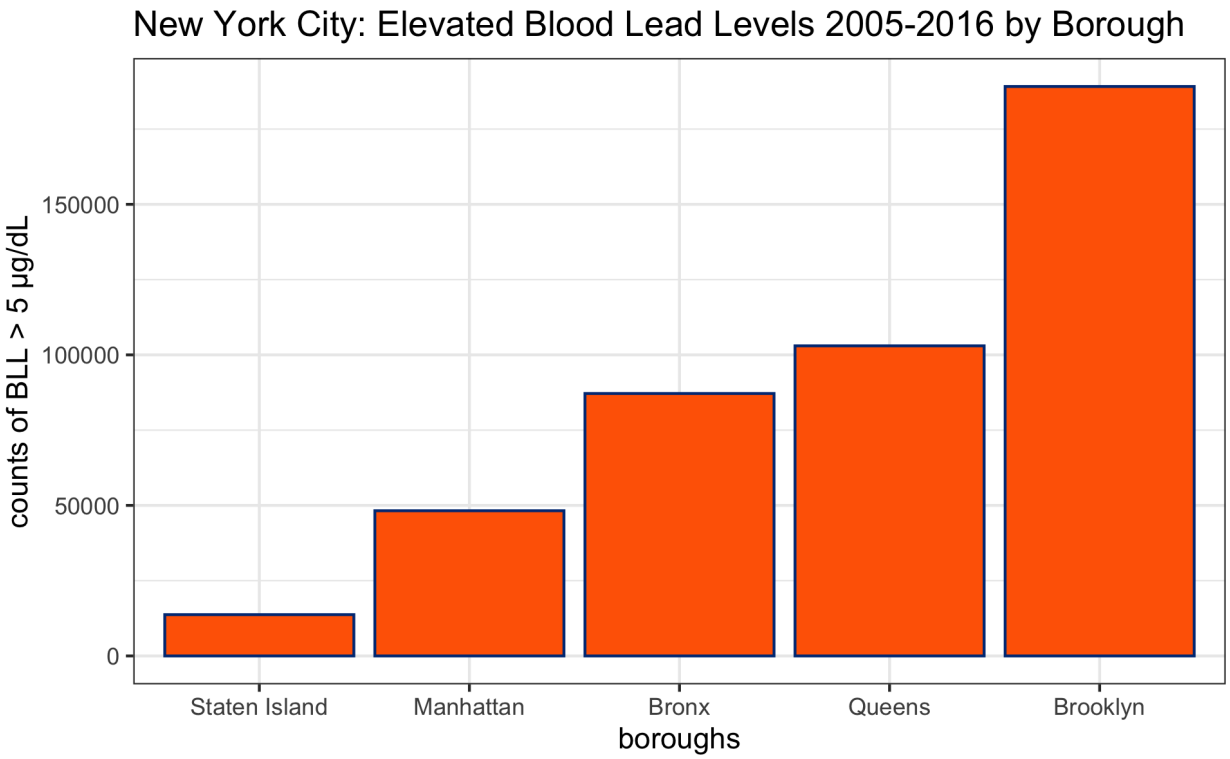
Question 5

Replicate the following bar chart. Since we want the graph to have an ascending order, we will need to factor borough_id with the levels in a different order than the default. Note that this graph covers the whole time period from the original dataset!

Here are the HEX codes used for the colors:

- #ff6600: orange
- #003884: blue

```
knitr::include_graphics('data/question_2_bar.png')
```



First, summarize the original dataset.

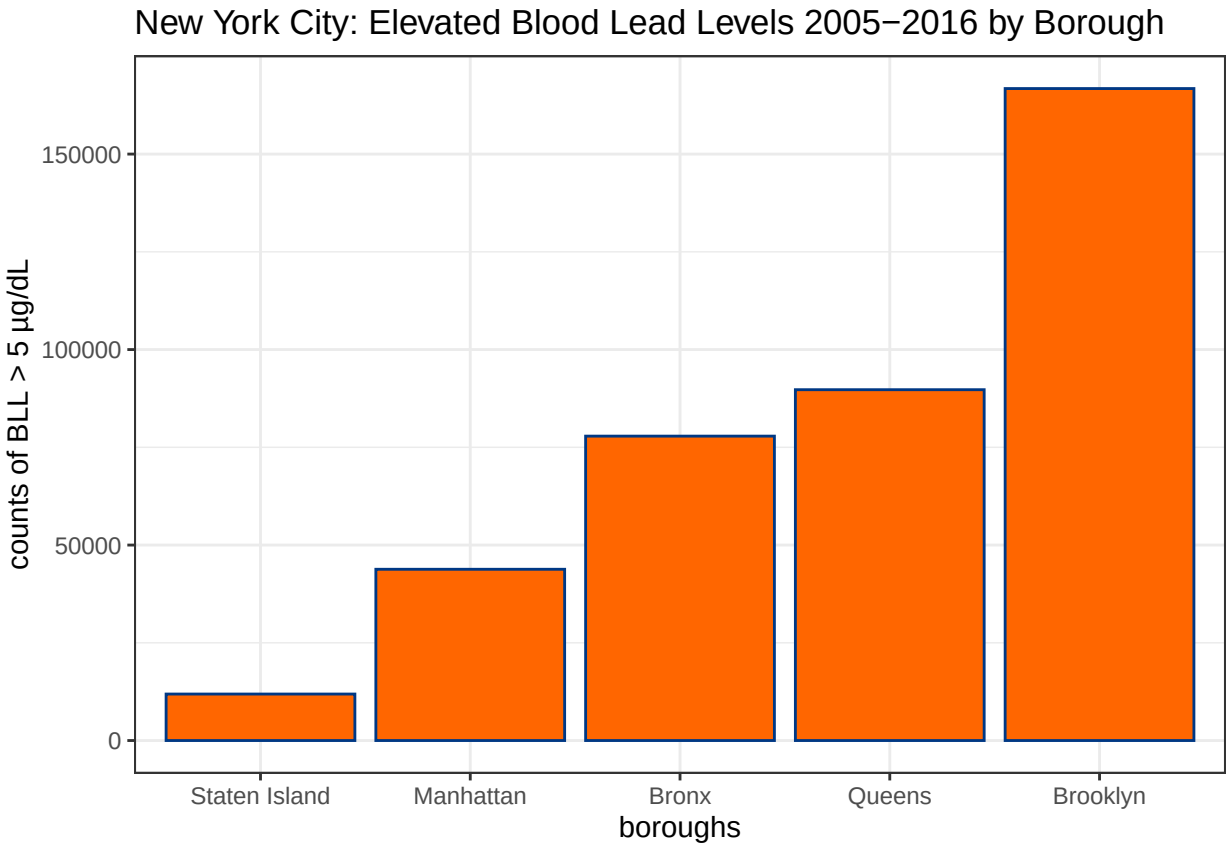
```
# your code here
# summarize all BLL > 5 in each borough
bll_nyc_bar <- bll_nyc %>%
  group_by(borough_id) %>%
  summarise(bll_5 = sum(bll_5))

# change the order of factor
bll_nyc_bar$borough_id <- factor(bll_nyc_bar$borough_id,
                                 levels = c(5,3,1,4,2), #"2","3","1","4","5"
                                 labels = c("Staten Island","Manhattan",
                                            "Bronx", "Queens", "Brooklyn"),
                                 ordered=TRUE)
```

Then make the graph!

```r
# NOTE: The graph image we asked you to replicate was created by adding together
# the number of kids with BLL > 5mcg/dL, BLL > 10, and BLL > 15, which doesn't
# make sense because any individual with BLL > 15 or BLL > 10 also has BLL > 5.
# By adding these together we double or tripled counted individuals.
# The code below makes the correct graph, which doesn't overcount people.

ggplot(bll_nyc_bar, aes(x = reorder(borough_id,bll_5), y = bll_5)) +
  geom_col(fill = "#ff6600", color = "#003884") +
  labs(x = "boroughs",
       y = "counts of BLL > 5 µg/dL",
       title = "New York City: Elevated Blood Lead Levels 2005-2016 by Borough") +
  theme_bw()
```



You're done! Please knit to pdf and upload to gradescope.