Question 1

```r
rm(list = ls())
library(tidyverse)
library(gapminder)
# use ?gapminder get the desciption of the dataset `gapminder`
```

Consider the dataset `gapminder` .

(a) Modify the `continent` factor by classifying the Americas' countries into `South America` and `North America`

```
Hint: the following countries are in South America.
```r
c("Argentina", "Bolivia", "Brazil", "Chile", "Colombia", "Ecuador", "Paraguay", "Peru",
"Trinidad and Tobago", "Uruguay", "Venezuela")
```
```

```r
# Examining the values of `continent`
unique(gapminder$continent)
```

```
## [1] Asia     Europe   Africa   Americas Oceania
## Levels: Africa Americas Asia Europe Oceania
```

```r
# Countries to be reassigned to `South America`
sAmerica = c("Argentina", "Bolivia", "Brazil", "Chile", "Colombia", "Ecuador", "Paragua
y", "Peru", "Trinidad and Tobago", "Uruguay", "Venezuela")

# Examining the current countries labelled `Americas`
gapminder %>%
  filter(continent == "Americas") %>%
  distinct(continent, country)
```

```
## # A tibble: 25 x 2
##    continent country
##    <fct>     <fct>
##  1 Americas  Argentina
##  2 Americas  Bolivia
##  3 Americas  Brazil
##  4 Americas  Canada
##  5 Americas  Chile
##  6 Americas  Colombia
##  7 Americas  Costa Rica
##  8 Americas  Cuba
##  9 Americas  Dominican Republic
## 10 Americas  Ecuador
## # … with 15 more rows
```

```r
# Relabelling the `Americas` values while leaving the other `continent` values unchanged
gapminder <- gapminder %>%
  mutate(continent = case_when(
    country %in% sAmerica ~ "South America",
    continent == "Americas" ~ "North America",
    TRUE ~ as.character(continent)
  )
  )

# Ensuring proper reclassification and no remaining `Americas` values
gapminder %>%
  filter(continent == "North America" | continent == "South America") %>%
  distinct(continent, country)
```

```
## # A tibble: 25 x 2
##    continent     country
##    <chr>         <fct>
##  1 South America Argentina
##  2 South America Bolivia
##  3 South America Brazil
##  4 North America Canada
##  5 South America Chile
##  6 South America Colombia
##  7 North America Costa Rica
##  8 North America Cuba
##  9 North America Dominican Republic
## 10 South America Ecuador
## # … with 15 more rows
```

```r
gapminder %>%
  filter(continent == "Americas")
```

```
## # A tibble: 0 x 6
## # … with 6 variables: country <fct>, continent <chr>, year <int>,
## #   lifeExp <dbl>, pop <int>, gdpPercap <dbl>
```

We see that there are no continent values for “Americas” remaining. Furthermore, the 25 countries originally labelled with a continent value of “Americas” have correctly been reclassified as either “South America” or “North America”.

In the following questions, use the dataset modified in (a).

Hint: you could use `case_when` function.

(b) How many countries are there in the dataset? How about for each continent?

```r
# Distinct country values in the `gapminder` dataset
n_distinct(gapminder$country)
```

```
## [1] 142
```

```
# Distinct countries for each continent
gapminder %>%
  group_by(continent) %>%
  summarise(country=n_distinct(country))
```

```
## # A tibble: 6 x 2
##   continent     country
##   <chr>           <int>
## 1 Africa             52
## 2 Asia               33
## 3 Europe             30
## 4 North America      14
## 5 Oceania             2
## 6 South America      11
```

```
# Comparing the number of countries we reassigned to `South America` and what we see in
 the summarized tibble
length(sAmerica)
```

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

```
# Showing the sum of summarized tibble is equal to the orignal 142 value we got at the b
eginning of the question
gapminder %>%
  group_by(continent) %>%
  summarise(country=n_distinct(country)) %>%
  summarise(sum=sum(country))
```

```
## # A tibble: 1 x 1
##     sum
##   <int>
## 1   142
```

There are 142 countries in the datset. There are 52 countries in Africa, 33 in Asia, 30 in Europe, 14 in North America, 2 in Oceania, and 11 in South America. Our South America figure can be double checked by examining the length of our “sAmerica” vector. We also find that the sum of countries on each continent (52+30+14+2+11) is equal to the total number of countries (142).

(c) For each year, which country had the largest gdp per capital?

```
gapminder %>%
  group_by(year) %>%
  top_n(n=1) %>%
  arrange(year) %>%
  select(country, year, gdpPercap)
```

```
## # A tibble: 12 x 3
## # Groups:   year [12]
##    country       year gdpPercap
##    <fct>        <int>     <dbl>
##  1 Kuwait        1952   108382.
##  2 Kuwait        1957   113523.
##  3 Kuwait        1962    95458.
##  4 Kuwait        1967    80895.
##  5 Kuwait        1972   109348.
##  6 Kuwait        1977    59265.
##  7 Saudi Arabia  1982    33693.
##  8 Norway        1987    31541.
##  9 Kuwait        1992    34933.
## 10 Norway        1997    41283.
## 11 Norway        2002    44684.
## 12 Norway        2007    49357.
```

Kuwait had the largest gdp per capital from 1952-1977 and again in 1992. Norway had the largest gdp per capital in 1987 and between 1997-2007. Saudi Arabia had the largest gdp per capital in 1982.

(d) For each continent, which country experienced the sharpest increment rate in life expectancy from 1997 to 2007?

```
# Since, after filtering, each country had two values (1997 & 2007). I used the lag betw
een the 2007 value and 1997 value and then divided by the 1997 value to find `inclifeExp
`.
# Note the 1997 rows would have irrelevant values for `inclifeExp`. This is handled by r
emoving the 1997 rows after creating the `inclifeExp` variable
gapminder %>%
  filter(year == 1997 | year == 2007) %>%
  mutate(inclifeExp = (lifeExp - lag(lifeExp))/lag(lifeExp)) %>%
  filter(year == 2007) %>%
  group_by(continent) %>%
  top_n(n=1) %>%
  select(country, continent, inclifeExp)
```

```
## # A tibble: 6 x 3
## # Groups:   continent [6]
##   country     continent     inclifeExp
##   <fct>       <chr>              <dbl>
## 1 Albania     Europe            0.0476
## 2 Bolivia     South America     0.0565
## 3 Haiti       North America     0.0749
## 4 New Zealand Oceania           0.0342
## 5 Rwanda      Africa            0.281
## 6 Yemen, Rep. Asia              0.0806
```

Albania had the sharpest increment rate in life expectancy from 1997 to 2007 in Europe with a value of 4.8%. Bolivia had the sharpest increment rate in life expectancy from 1997 to 2007 in South America with a value of 5.6%. New Zealand had the sharpest increment rate in life expectancy from 1997 to 2007 in Oceania with a value of 3.4%. Haiti had the sharpest increment rate in life expectancy from 1997 to 2007 in North America with a value of 7.5%. Rwanda had the sharpest increment rate in life expectancy from 1997 to 2007 in Africa with a value of 28.1%. Yemen had the sharpest increment rate in life expectancy from 1997 to 2007 in Asia with a value of 8.1%.

(e) Focus on the data in year 2007, what are the correlation coefficients between life expectancy and gdp per capital for each continent?

```
# cor() find the correlation coefficient between two variables
gapminder %>%
  filter(year == 2007) %>%
  group_by(continent) %>%
  summarise(r = cor(lifeExp, gdpPercap))
```

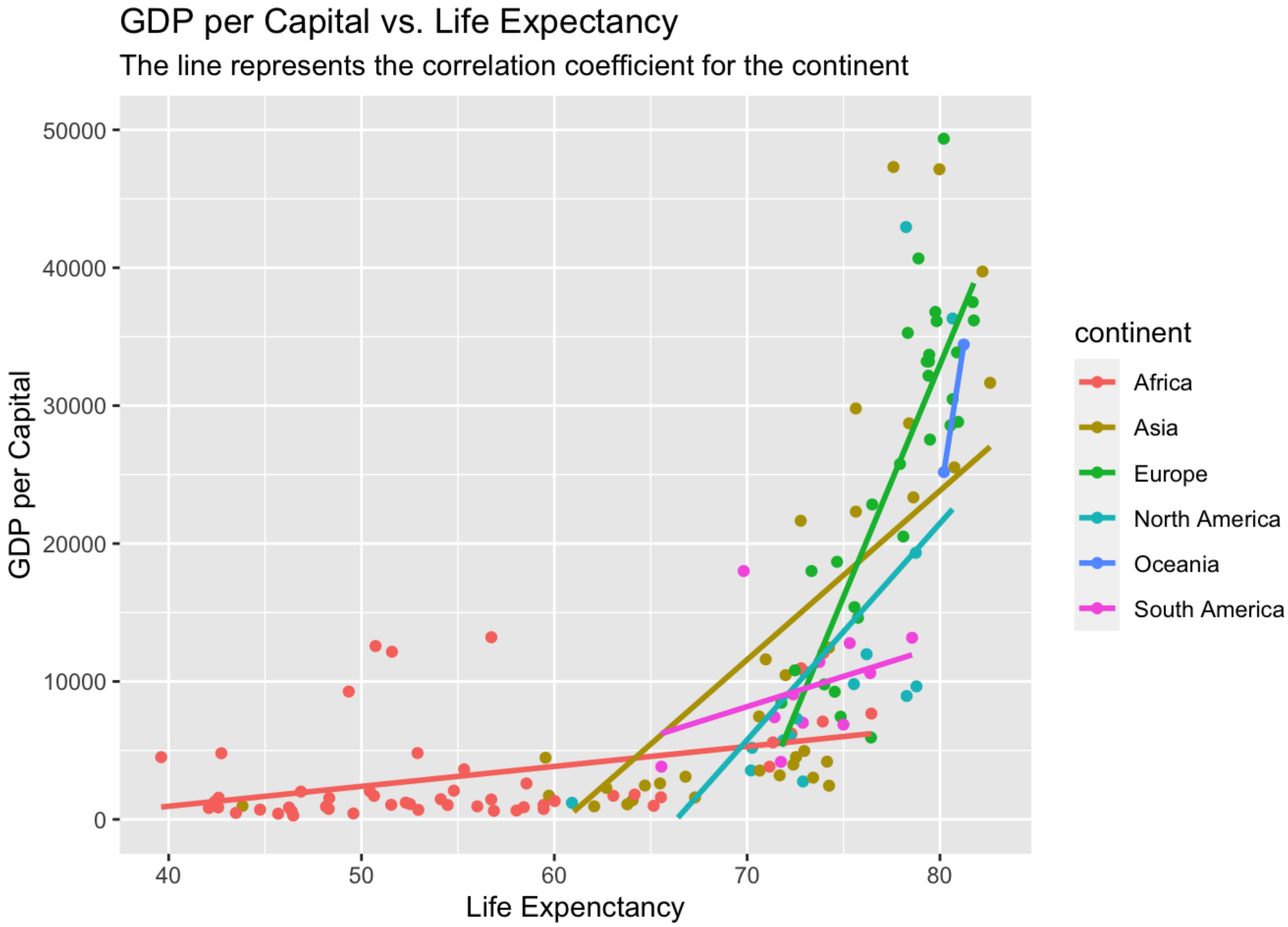
```
## # A tibble: 6 x 2
##   continent         r
##   <chr>         <dbl>
## 1 Africa        0.385
## 2 Asia          0.689
## 3 Europe        0.850
## 4 North America 0.645
## 5 Oceania       1
## 6 South America 0.362
```

Oceania boasts the largest correlation coefficient for 2007 data on life expectancy and gdp per capital with a value of 1.00. Note: This can be attributed to the small sample size of Oceania (2). Africa and South America return relatively low coefficients with .3847 and .3619, respectively. Europe shows a strong correlation with a value of .8500. Asia and America both show modernately strong positive coefficients being .6894 and .6447, respectively.

(f) Visualize part (e) by plotting gdp per capital vs life expectancy.

```
# visualization for part (e)
gapminder %>%
  filter(year == 2007) %>%
  drop_na() %>%
  ggplot(aes(x = lifeExp, y = gdpPercap, color = continent)) +
  geom_point() +
  labs(title = "GDP per Capital vs. Life Expectancy",subtitle = "The line represents the
correlation coefficient for the continent" ,x= "Life Expenctancy", y = "GDP per Capital"
) +
  geom_smooth(method='lm', formula= y~x, se = FALSE) +
  ylim(c(0,50000))
```

```
## Warning: Removed 57 rows containing missing values (geom_smooth).
```


GDP per Capital vs. Life Expectancy
The line represents the correlation coefficient for the continent
50000
40000
30000
20000
10000
0
GDP per Capital
40
50
60
70
80
Life Expenctancy
continent
Africa
Asia
Europe
North America
Oceania
South America

Question 2

Consider the `flights` dataset in the package `nycflights13`.

```
library(nycflights13)
library(tidyverse)
```

(a) Add a column that is the amount of time gained in the air (`gain = dep_delay - arr_delay`)

```
# Creating `gain` variable
flights <- flights %>%
  mutate(gain = dep_delay-arr_delay)
```

(b) Sort part (a) descedingly by the column you just created. Store the result as `flights_gain`.

```
# `flights` tibble ordered by `gain` (descending)
flights_gain <- flights %>%
  arrange(desc(gain))
head(flights_gain)
```

```
## # A tibble: 6 x 20
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     6    13     1907           1512       235     2134           1928
## 2  2013     2    26     1000            900        60     1513           1540
## 3  2013     2    23     1226            900       206     1746           1540
## 4  2013     5    13     1917           1900        17     2149           2251
## 5  2013     2    27      924            900        24     1448           1540
## 6  2013     7    14     1917           1829        48     2109           2135
## # … with 12 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>, gain <dbl>
```

(c) On average, did flights gain or lose time? (Hint: not average gain, but as percentage of positive gain.)

```
# Finding percentage of flights with positive gain
flights_gain %>%
  drop_na() %>%
  count(gain>0) %>%
  mutate(freq = n / sum(n))
```

```
## # A tibble: 2 x 3
##   `gain > 0`      n  freq
##   <lgl>       <int> <dbl>
## 1 FALSE      105781 0.323
## 2 TRUE       221565 0.677
```

67.7% of flgiths gained time, so on average flights gained time.

(d) On average, did flights heading to SeaTac ("SEA") gain or loose time?

```
# Same as (c), but with the subset of flights heading to SeaTac ("SEA")
flights_gain %>%
  filter(dest == "SEA") %>%
  drop_na() %>%
  count(gain>0) %>%
  mutate(freq = n / sum(n))
```

```
## # A tibble: 2 x 3
##   `gain > 0`     n  freq
##   <lgl>      <int> <dbl>
## 1 FALSE        929 0.239
## 2 TRUE        2956 0.761
```

76.1% of flights headed to SeaTac gained time, so on average flights gained time.

(e) Summerize the mean, min and max of the `air_time` column for flights from `JFK` to `SEA`.

```
# mean(), min(), and max() of Flights with origin "JFK" and destination "SEA"
# max() or min() value of `origin` and `dest` will always be "JFK" and "SEA", respective
ly, as those are the only values represented in those variables.
flights %>%
  filter(origin == "JFK" & dest == "SEA") %>%
  summarise(origin=max(origin),
            dest=max(dest),
            mean = mean(air_time, na.rm = TRUE),
            min = min(air_time, na.rm = TRUE),
            max = max(air_time, na.rm = TRUE)
            )
```

```
## # A tibble: 1 x 5
##   origin dest   mean   min   max
##   <chr>  <chr> <dbl> <dbl> <dbl>
## 1 JFK    SEA    329.   275   389
```

Flights from "JFK" to "SEA" had a mean airtime of 329 minutes, a minimum value of 275 minutes, and a maximum value of 389 minutes.

(f) In which month was the average departure delay the greatest?

```
flights %>%
  group_by(month) %>%
  summarise(avg_delay = mean(dep_delay, na.rm = TRUE)) %>%
  top_n(n=1)
```

```
## # A tibble: 1 x 2
##   month avg_delay
##   <int>     <dbl>
## 1     7      21.7
```

July has the greatest average departure delay. This can be assumed to be due to the increase in air traffic during summer travel.

(g) In which airport were the average arrival delays the highest?

```
flights %>%
  group_by(dest) %>%
  summarise(avg_arr_delay = mean(arr_delay, na.rm = TRUE)) %>%
  top_n(n=1)
```

```
## # A tibble: 1 x 2
##   dest  avg_arr_delay
##   <chr>         <dbl>
## 1 CAE            41.8
```

CAE has the highest average arrival delays.

(h) Which city was flown to with the highest average speed?

```
flights %>%
  mutate(speed = distance / air_time) %>%
  group_by(dest) %>%
  summarise(avg_speed = mean(speed, na.rm = TRUE)) %>%
  top_n(n=1)
```

```
## # A tibble: 1 x 2
##   dest  avg_speed
##   <chr>     <dbl>
## 1 ANC        8.17
```

ANC was flown to with the higest average speed with a value of 8.16 miles per minute (equivalent to 489.6 miles per hour or 787.9 kilometers per hour).

(i) Create a data frame of the average arrival delay for each destination, then use `left_join` to join on the `airports` dataframe, which has the airport info. (Hint: read the documentation of `airports` for the airport codes.)

```
# Creating data frame for average arrival delay for each destination
avg_arrival_delay <- flights %>%
  group_by(dest) %>%
  summarise(avg_arr_delay = mean(arr_delay, na.rm = TRUE))

# Left joining with `airports` to create a data frame with additional information about
 our destination airports
left_join(avg_arrival_delay, airports, by = c("dest" = "faa"))
```

```
## # A tibble: 105 x 9
##    dest  avg_arr_delay name                lat    lon   alt    tz dst   tzone
##    <chr>         <dbl> <chr>             <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 ABQ            4.38 Albuquerque Int…   35.0 -107.   5355    -7 A     America/…
##  2 ACK            4.85 Nantucket Mem      41.3  -70.1    48    -5 A     America/…
##  3 ALB           14.4  Albany Intl        42.7  -73.8   285    -5 A     America/…
##  4 ANC           -2.5  Ted Stevens Anc…   61.2 -150.    152    -9 A     America/…
##  5 ATL           11.3  Hartsfield Jack…   33.6  -84.4  1026    -5 A     America/…
##  6 AUS            6.02 Austin Bergstro…   30.2  -97.7   542    -6 A     America/…
##  7 AVL            8.00 Asheville Regio…   35.4  -82.5  2165    -5 A     America/…
##  8 BDL            7.05 Bradley Intl       41.9  -72.7   173    -5 A     America/…
##  9 BGR            8.03 Bangor Intl        44.8  -68.8   192    -5 A     America/…
## 10 BHM           16.9  Birmingham Intl    33.6  -86.8   644    -6 A     America/…
## # … with 95 more rows
```

Question 3

(a) There is a csv file called `groceries.csv` in this directory. Read the csv file using `read_csv` from `tidyverse` and store the data frame as `groceries`. The datset shows the prices of some common groceries item in 4 different stores.

```
library(tidyverse)
groceries <- read_csv("groceries.csv")
groceries
```

```
## # A tibble: 10 x 5
##    groceries         storeA storeB storeC storeD
##    <chr>              <dbl>  <dbl>  <dbl>  <dbl>
##  1 lettuce             1.17   1.78   1.29   1.29
##  2 potatoes            1.77   1.98   1.99   1.99
##  3 milk                1.49   1.69   1.79   1.59
##  4 eggs                0.65   0.99   0.69   1.09
##  5 bread               1.58   1.7    1.89   1.89
##  6 cereal              3.13   3.15   2.99   3.09
##  7 ground.beef         2.09   1.88   2.09   2.49
##  8 tomato.soup         0.62   0.65   0.65   0.69
##  9 laundry.detergent   5.89   5.99   5.99   6.99
## 10 aspirin             4.46   4.84   4.99   5.15
```

The table shows the prices of different items in 4 different stores.

(b) Is the data frame in wide format or long format?

The data frame is in wide format. We can justify this by the fact if we continue to add more stores to the data frame we would be creating more columns, and as a result a wider data frame. A long format would be if there was 4 rows for each item and `store` was a column with values {A,B,C,D} and a column for `price` corresponding to the price for that item at that store.

(c) Try to convert it into the other format. Store it as `groceries2`.

```
# Converting groceries dataset to long format using pivot_long(). The result is a 40x3 d
ata frame.
groceries2 <- groceries %>%
  pivot_longer(-groceries, names_to = "store", names_prefix = "store", values_to = "pric
e")
groceries2
```

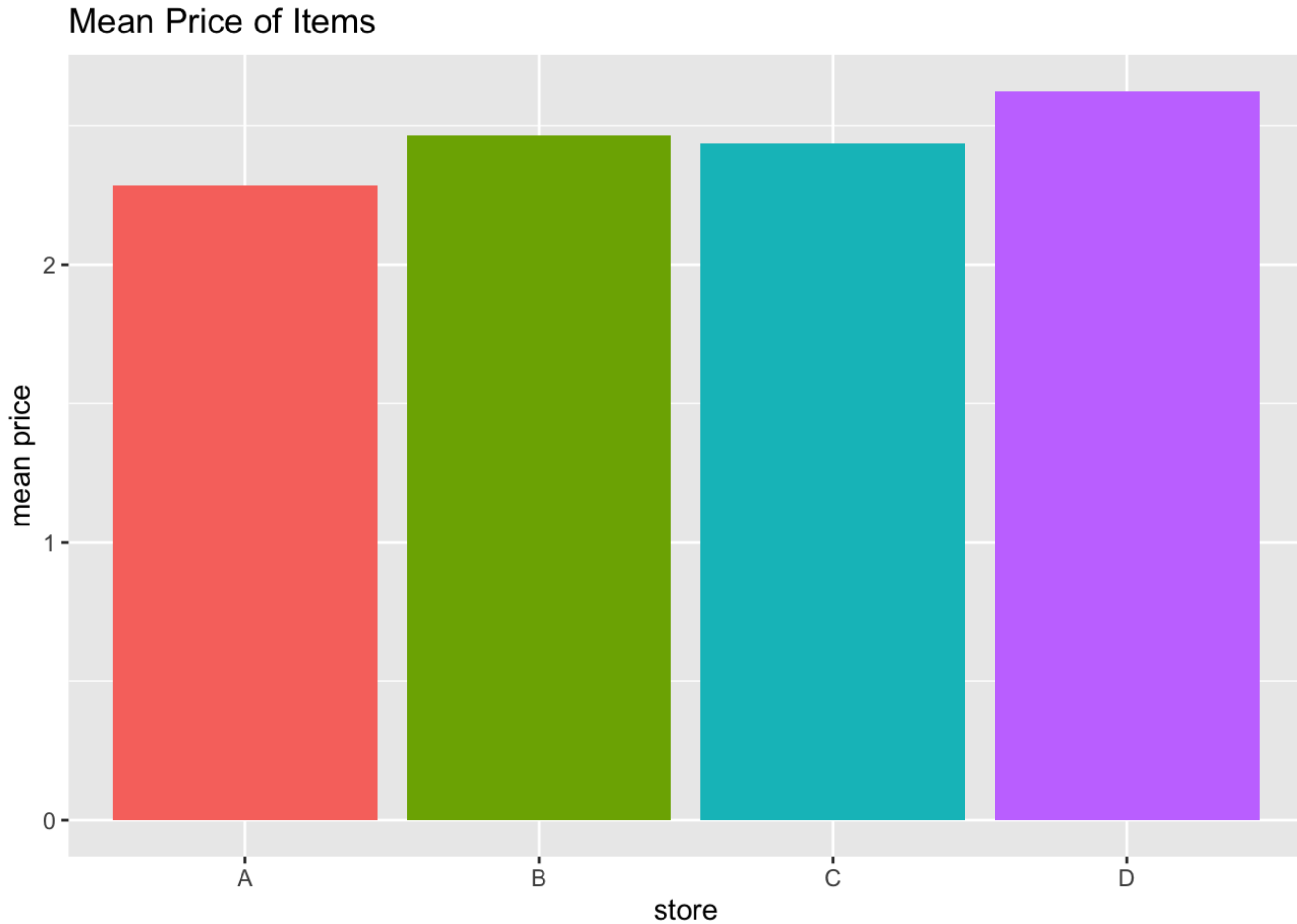
```
## # A tibble: 40 x 3
##    groceries store price
##    <chr>     <chr> <dbl>
##  1 lettuce   A      1.17
##  2 lettuce   B      1.78
##  3 lettuce   C      1.29
##  4 lettuce   D      1.29
##  5 potatoes  A      1.77
##  6 potatoes  B      1.98
##  7 potatoes  C      1.99
##  8 potatoes  D      1.99
##  9 milk      A      1.49
## 10 milk      B      1.69
## # … with 30 more rows
```

(d) Use a randomized block design to analysis the store prices. Is there a store marking up the item prices?

```
# http://www.r-tutor.com/elementary-statistics/analysis-variance/randomized-block-design
prices = c(t(as.matrix(groceries[-1])))
stores = c("StoreA", "StoreB", "StoreC", "StoreD")
num_factors = 4
num_blocks = 10
treatment_factors = gl(num_factors, 1, num_blocks*num_factors, factor(stores))
block_factors = gl(num_blocks, num_factors, num_factors*num_blocks)
av = aov(prices ~ treatment_factors + block_factors)
summary(av)
```

```
##                   Df Sum Sq Mean Sq F value Pr(>F)
## treatment_factors  3   0.59   0.195   4.344 0.0127 *
## block_factors      9 115.19  12.799 284.722 <2e-16 ***
## Residuals         27   1.21   0.045
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
groceries2 %>%
  group_by(store) %>%
  summarise(price = mean(price)) %>%
  ggplot(aes(store, price, fill = store)) +
  geom_col() +
  theme(legend.position = "none") +
  labs(title = "Mean Price of Items",y = "mean price")
```



From our randomized block design analysis, with a p-value of .0127 that is less than our significance value of .05, we reject the null hypothesis that the mean sales price of the four stores are equal. It appears, from our bar chart, as though store D is marking up prices. In the opposite direction, store A appears to be the best for any bargain shopper.