Project 2 DATA-607

Kenan Sooklall

3/5/2021

```
#library(dt)
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

For this project I will be cleaning and visualize 3 different datasets

Country prision admissions

Clean country_prision_admissions.csv data set This dataset has 16 columns and 3143 rows

```
pdf <- read.csv('/home/kenan/Documents/learning/masters/CUNY-SPS-Masters-DS/DATA_607/projects/project_2/

glimpse(pdf)
```

```
## Rows: 3,143
## Columns: 16
## $ fips             <int> 1001, 1003, 1005, 1007, 1009, 1011, 1013, 1015, 1017,~
## $ county           <chr> "Autauga County", "Baldwin County", "Barbour County",~
## $ state            <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL",~
## $ admitsPer10k2006 <dbl> 44.25665, 24.63739, 75.39988, 21.97416, 16.13939, 66.~
## $ admitsPer10k2013 <dbl> 19.225189, 17.703371, 10.378827, 11.109136, 11.781012~
## $ admitsPer10k2014 <dbl> 18.593736, 16.540820, 12.273589, 6.664889, 6.930127, ~
```

```
## $ valid06          <chr> "true", "true", "true", "true", "true", "true", "true~
## $ valid13          <chr> "true", "true", "true", "true", "true", "true", "true~
## $ valid14          <chr> "true", "true", "true", "true", "true", "", "true", "~
## $ population2006   <int> 51328, 168121, 27861, 22099, 55485, 10776, 20815, 115~
## $ population2013   <int> 55136, 195443, 26978, 22504, 57720, 10605, 20289, 116~
## $ population2014   <int> 55395, 200111, 26887, 22506, 57719, 10764, 20296, 115~
## $ admissions2006   <int> 243, 461, 206, 50, 93, 71, 75, 542, 173, 65, 189, 22,~
## $ admissions2013   <int> 106, 346, 28, 25, 68, 9, 76, 430, 84, 100, 103, 18, 7~
## $ admissions2014   <chr> "103", "331", "33", "15", "40", NA, "70", "430", "88"~
## $ source           <chr> "NCRP", "NCRP", "NCRP", "NCRP", "NCRP", "NCRP", "NCRP~
```

I am only interested in the admission per 10k for the years os 2006, 2013 and 2014. Also the format of the data need to change since I want all the years in one column and their corresponding counts in another. I will also rename the column names to something more appropriate.

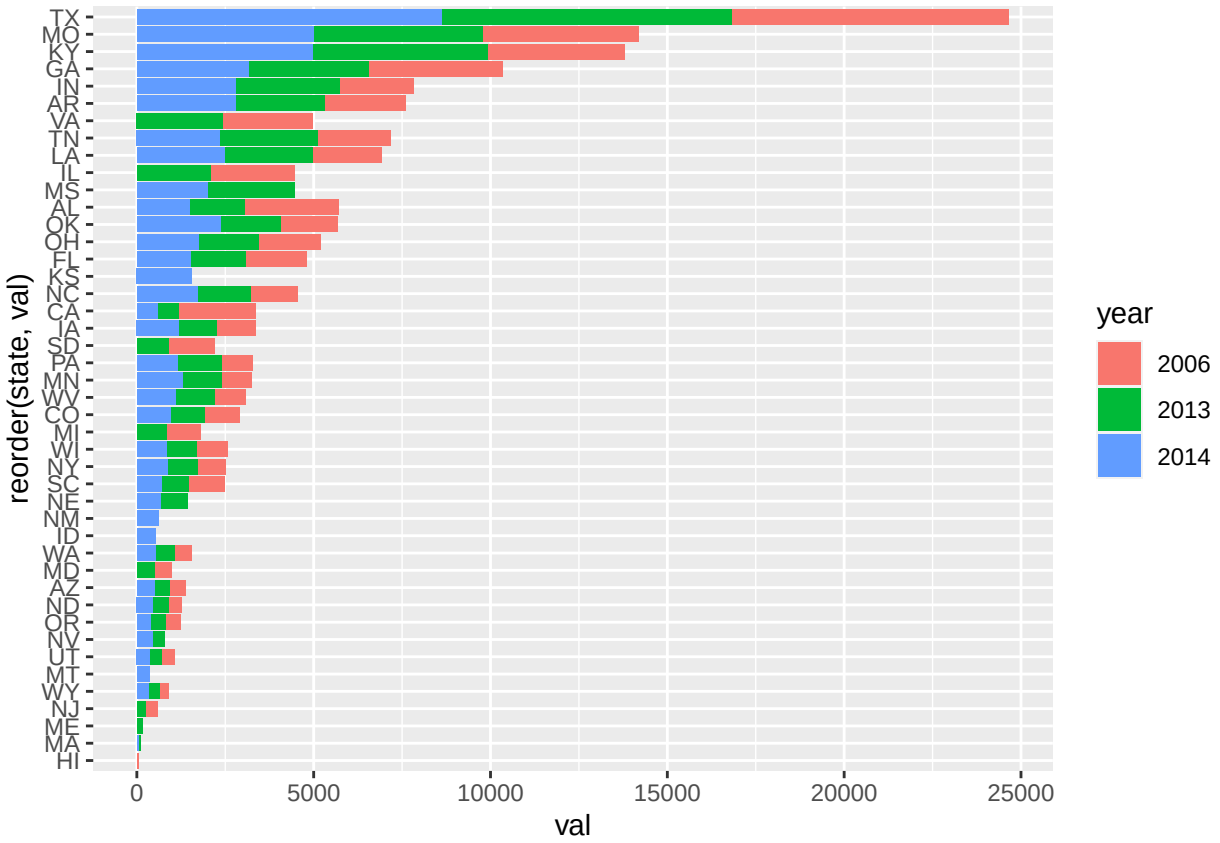
```
cols= c('county', 'state', 'admitsPer10k2006', 'admitsPer10k2013', 'admitsPer10k2014')

pdf <- pdf[,cols]
pdf <- pdf %>% rename('2006'='admitsPer10k2006', '2013'='admitsPer10k2013', '2014'='admitsPer10k2014') %

glimpse(pdf)
```

```
## Rows: 7,492
## Columns: 4
## $ county <chr> "Autauga County", "Autauga County", "Autauga County", "Baldwin ~
## $ state  <chr> "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL", "AL~
## $ year   <chr> "2006", "2013", "2014", "2006", "2013", "2014", "2006", "2013",~
## $ count  <dbl> 44.256652, 19.225189, 18.593736, 24.637387, 17.703371, 16.54082~
```

With the dataset cleaned I can now perform EDA

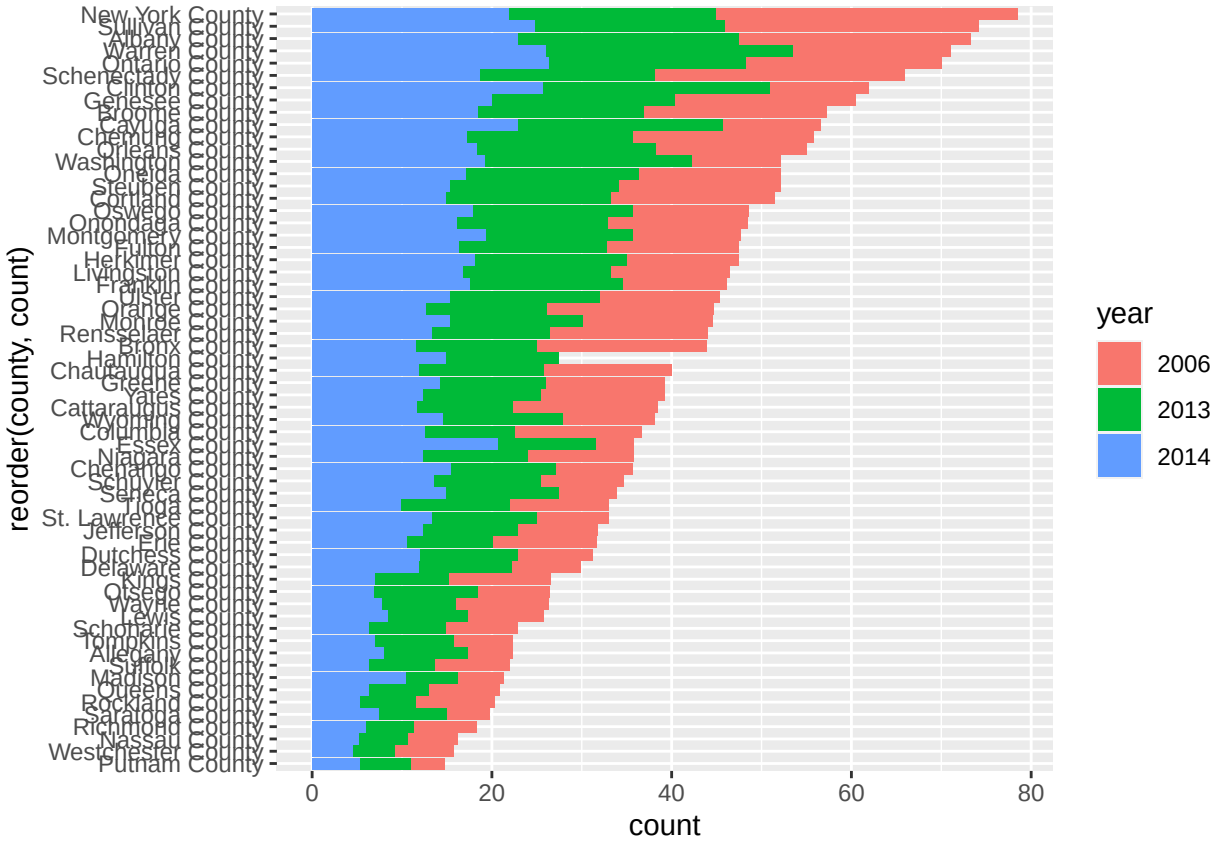
```
pdf %>% group_by(state, year) %>% summarise(val=sum(count)) %>% ggplot(aes(x=reorder(state, val), y=val
```

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

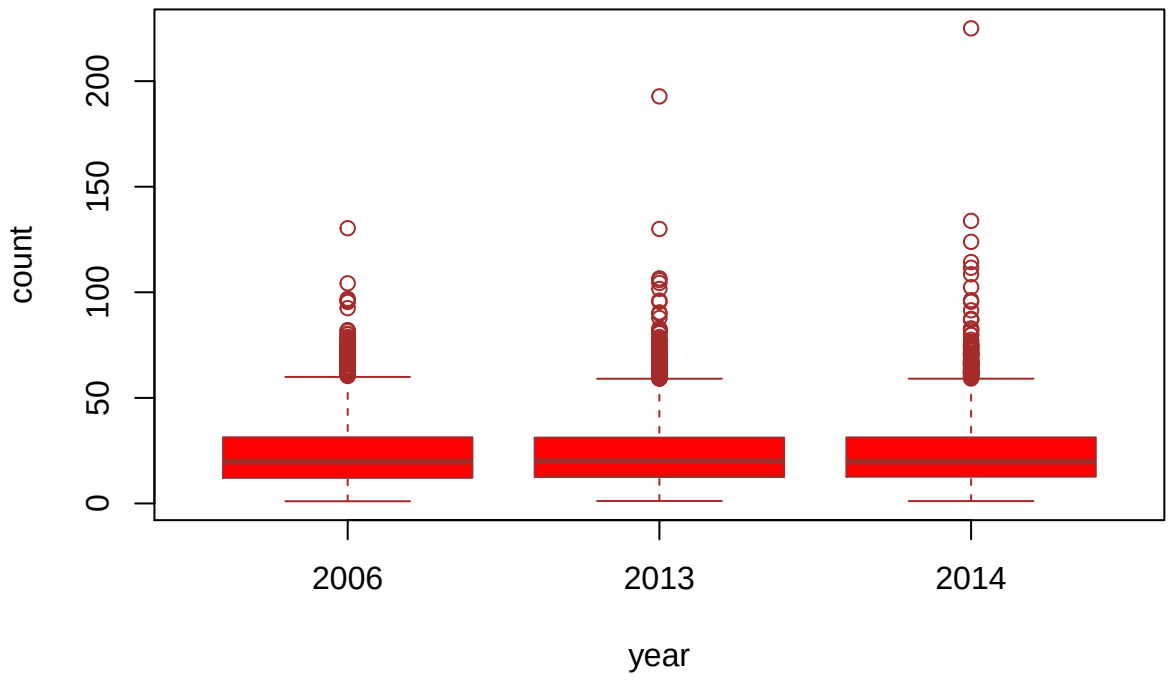


Wow Texas wins by a landslide, it seems like if you have a pulse in Texas you qualify for some prison sentence.

```
pdf %>% filter(state == 'NY') %>% ggplot(aes(x=count, y=reorder(county, count), fill=year)) + geom_col()
```




```
boxplot(count~year, data=pdf, col='red', border='brown')
```



For just NY these box plot show more there are outlier from 2006 to 2014

County dataset

The county dataset has 37 columns and 3220 rows, this dataset has a lot of grainular information for race to income to commute and many more.

```
cdf <- read.csv('/home/kenan/Documents/learning/masters/CUNY-SPS-Masters-DS/DATA_607/projects/project_2/

glimpse(cdf)
```

```
## Rows: 3,220
## Columns: 37
## $ CountyId         <int> 1001, 1003, 1005, 1007, 1009, 1011, 1013, 1015, 1017,~
## $ State            <chr> "Alabama", "Alabama", "Alabama", "Alabama", "Alabama"~
## $ County           <chr> "Autauga County", "Baldwin County", "Barbour County",~
## $ TotalPop         <int> 55036, 203360, 26201, 22580, 57667, 10478, 20126, 115~
## $ Men              <int> 26899, 99527, 13976, 12251, 28490, 5616, 9416, 55593,~
## $ Women            <int> 28137, 103833, 12225, 10329, 29177, 4862, 10710, 5993~
## $ Hispanic         <dbl> 2.7, 4.4, 4.2, 2.4, 9.0, 0.3, 0.3, 3.6, 2.2, 1.6, 7.7~
## $ White            <dbl> 75.4, 83.1, 45.7, 74.6, 87.4, 21.6, 52.2, 72.7, 56.2,~
## $ Black            <dbl> 18.9, 9.5, 47.8, 22.0, 1.5, 75.6, 44.7, 20.4, 39.3, 5~
## $ Native           <dbl> 0.3, 0.8, 0.2, 0.4, 0.3, 1.0, 0.1, 0.2, 0.3, 0.5, 0.4~
## $ Asian            <dbl> 0.9, 0.7, 0.6, 0.0, 0.1, 0.7, 1.1, 1.0, 1.0, 0.1, 0.4~
## $ Pacific          <dbl> 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0~
```

```
## $ VotingAgeCitizen <int> 41016, 155376, 20269, 17662, 42513, 8212, 15459, 8838~
## $ Income           <int> 55317, 52562, 33368, 43404, 47412, 29655, 36326, 4368~
## $ IncomeErr        <int> 2838, 1348, 2551, 3431, 2630, 5376, 2701, 1491, 2011,~
## $ IncomePerCap     <int> 27824, 29364, 17561, 20911, 22021, 20856, 19004, 2363~
## $ IncomePerCapErr  <int> 2024, 735, 798, 1889, 850, 2355, 943, 793, 1205, 1354~
## $ Poverty          <dbl> 13.7, 11.8, 27.2, 15.2, 15.6, 28.5, 24.4, 18.6, 18.8,~
## $ ChildPoverty     <dbl> 20.1, 16.1, 44.9, 26.6, 25.4, 50.4, 34.8, 26.6, 29.1,~
## $ Professional     <dbl> 35.3, 35.7, 25.0, 24.4, 28.5, 19.7, 26.9, 29.0, 24.3,~
## $ Service          <dbl> 18.0, 18.2, 16.8, 17.6, 12.9, 17.1, 17.3, 17.5, 13.5,~
## $ Office           <dbl> 23.2, 25.6, 22.6, 19.7, 23.3, 18.6, 18.5, 23.7, 23.0,~
## $ Construction     <dbl> 8.1, 9.7, 11.5, 15.9, 15.8, 14.0, 11.6, 10.4, 11.6, 1~
## $ Production       <dbl> 15.4, 10.8, 24.1, 22.4, 19.5, 30.6, 25.7, 19.4, 27.6,~
## $ Drive            <dbl> 86.0, 84.7, 83.4, 86.4, 86.8, 73.1, 83.6, 85.0, 87.1,~
## $ Carpool          <dbl> 9.6, 7.6, 11.1, 9.5, 10.2, 15.7, 12.6, 9.2, 9.7, 12.1~
## $ Transit          <dbl> 0.1, 0.1, 0.3, 0.7, 0.1, 0.3, 0.0, 0.2, 0.2, 0.4, 0.1~
## $ Walk             <dbl> 0.6, 0.8, 2.2, 0.3, 0.4, 6.2, 0.9, 1.3, 0.6, 0.3, 0.6~
## $ OtherTransp      <dbl> 1.3, 1.1, 1.7, 1.7, 0.4, 1.7, 0.9, 1.1, 0.5, 0.3, 1.8~
## $ WorkAtHome       <dbl> 2.5, 5.6, 1.3, 1.5, 2.1, 3.0, 2.0, 3.2, 2.0, 2.0, 1.7~
## $ MeanCommute      <dbl> 25.8, 27.0, 23.4, 30.0, 35.0, 29.8, 23.2, 24.8, 23.6,~
## $ Employed         <int> 24112, 89527, 8878, 8171, 21380, 4290, 7727, 47392, 1~
## $ PrivateWork      <dbl> 74.1, 80.7, 74.1, 76.0, 83.9, 81.4, 79.1, 74.9, 84.5,~
## $ PublicWork       <dbl> 20.2, 12.9, 19.1, 17.4, 11.9, 13.6, 15.3, 19.9, 11.8,~
## $ SelfEmployed     <dbl> 5.6, 6.3, 6.5, 6.3, 4.0, 5.0, 5.3, 5.1, 3.7, 8.1, 4.5~
## $ FamilyWork       <dbl> 0.1, 0.1, 0.3, 0.3, 0.1, 0.0, 0.3, 0.1, 0.0, 0.0, 0.4~
## $ Unemployment     <dbl> 5.2, 5.5, 12.4, 8.2, 4.9, 12.1, 7.6, 10.1, 6.4, 5.3, ~
```

Given the side of this dataset I would first filter for NY and drop a lot of the columns I won't be using and rows that contain NA.

```
usecols = c('County', 'IncomePerCap', 'Unemployment', 'MeanCommute', 'WorkAtHome')
cdf <- cdf %>% filter(State == "New York") %>% select(usecols) %>% drop_na
```

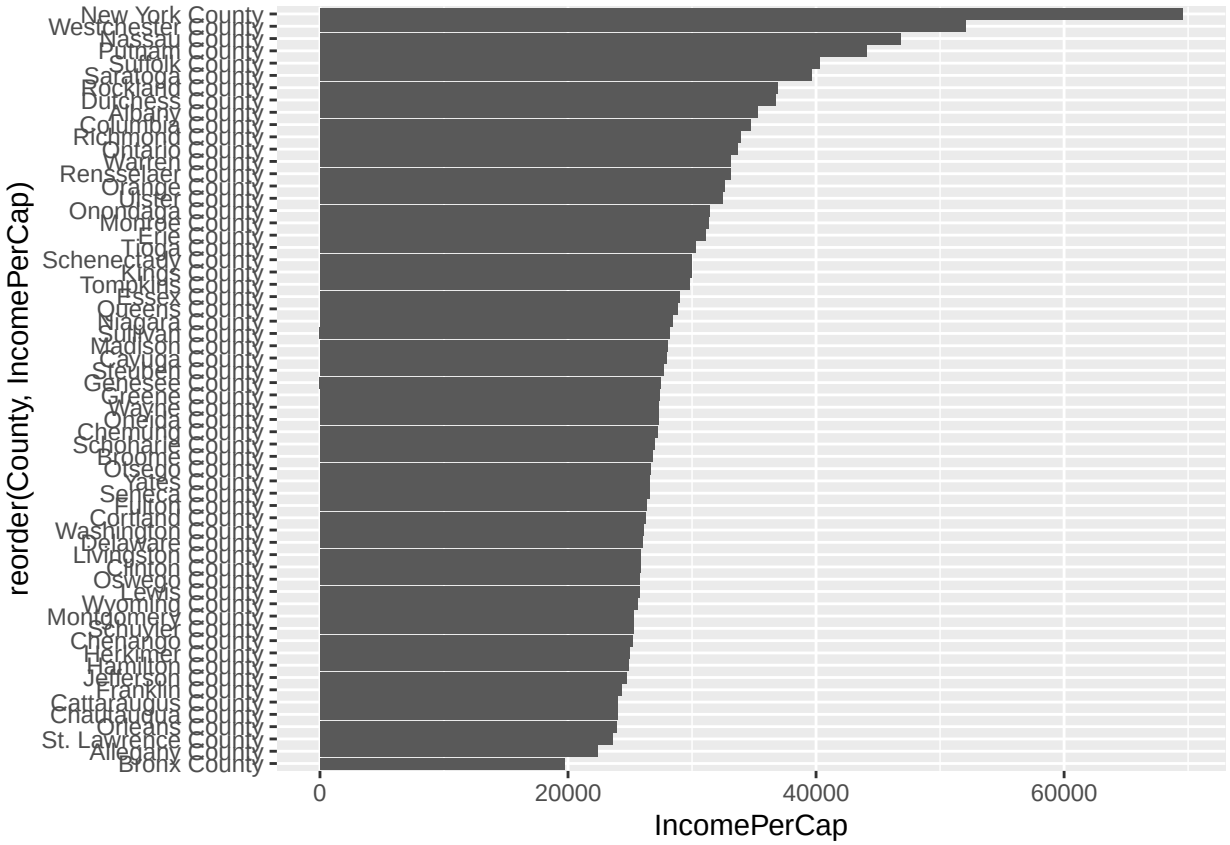
```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(usecols)` instead of `usecols` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
glimpse(cdf)
```

```
## Rows: 62
## Columns: 5
## $ County       <chr> "Albany County", "Allegany County", "Bronx County", "Broo~
## $ IncomePerCap <int> 35278, 22377, 19721, 26790, 23984, 27957, 23962, 27209, 2~
## $ Unemployment <dbl> 5.3, 7.3, 11.6, 7.2, 7.0, 5.9, 7.7, 5.3, 6.7, 5.8, 5.0, 4~
## $ MeanCommute  <dbl> 20.4, 21.4, 44.2, 19.7, 21.9, 23.0, 18.0, 19.6, 24.1, 19.~
## $ WorkAtHome   <dbl> 3.5, 2.8, 3.1, 3.1, 3.0, 3.4, 3.7, 2.3, 5.0, 2.7, 6.3, 4.~
```

Which country has the highest IncomePerCap, Unemployment?

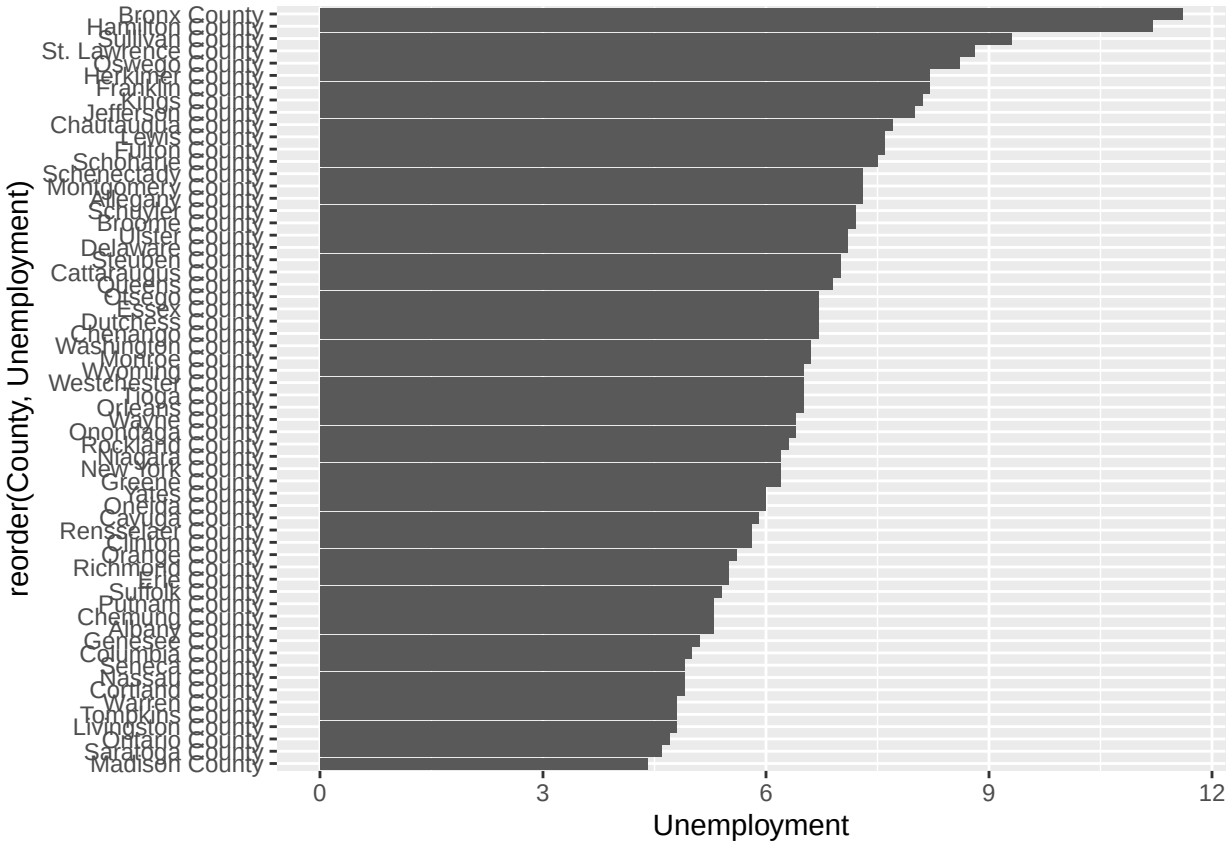
```
cdf %>% ggplot(aes(x=reorder(County, IncomePerCap), y=IncomePerCap)) + geom_col() + coord_flip()
```



It looks like New York County (Manhattan) has the highest Income per capital and Bronx County has the lowest

Which country has the highest Unemployment rate?

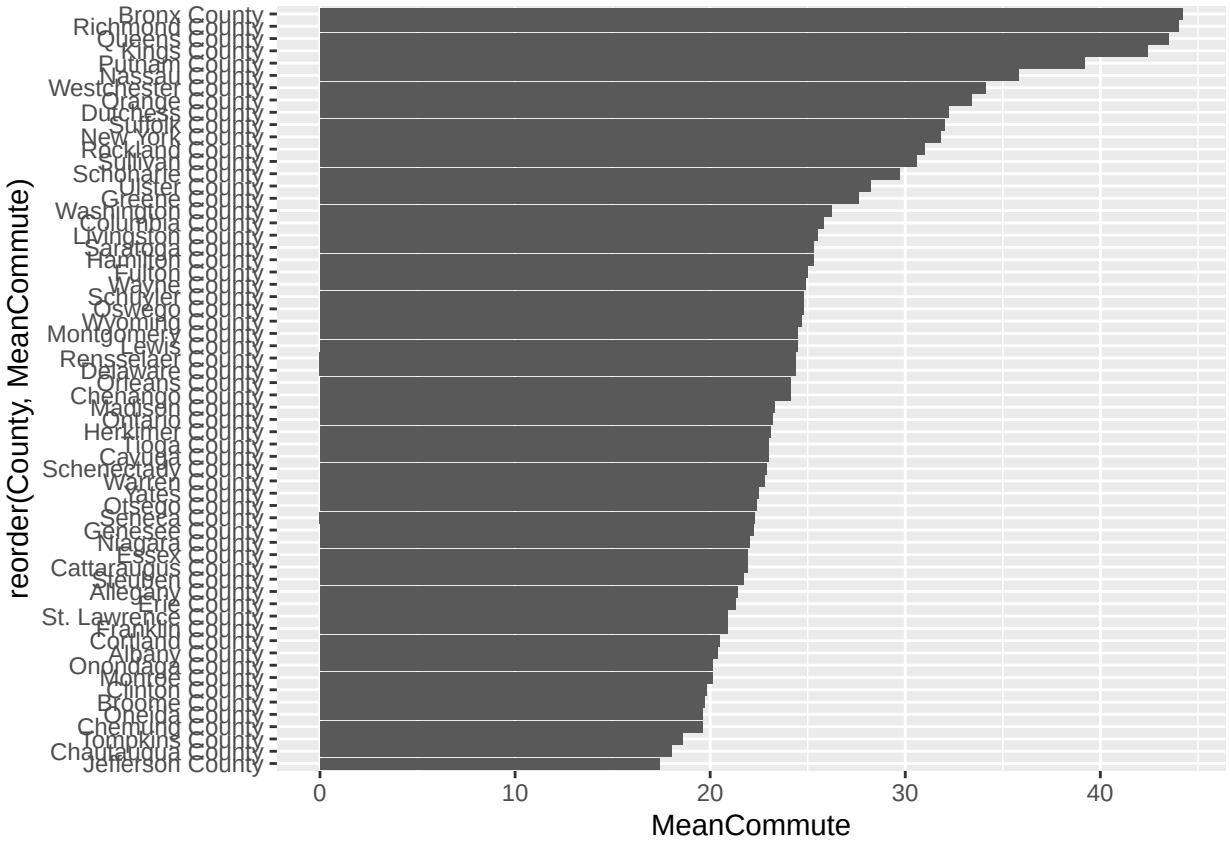
```
cdf %>% ggplot(aes(x=reorder(County, Unemployment), y=Unemployment)) + geom_col() + coord_flip()
```



From our first plot we saw Bronx County has the lowest income per capital so it makes sense that they has the highest unemployment rate. Surprisingly New York County doesn't have the lowerst unemployment, that goes to Madison County.

Which county commutes the most?

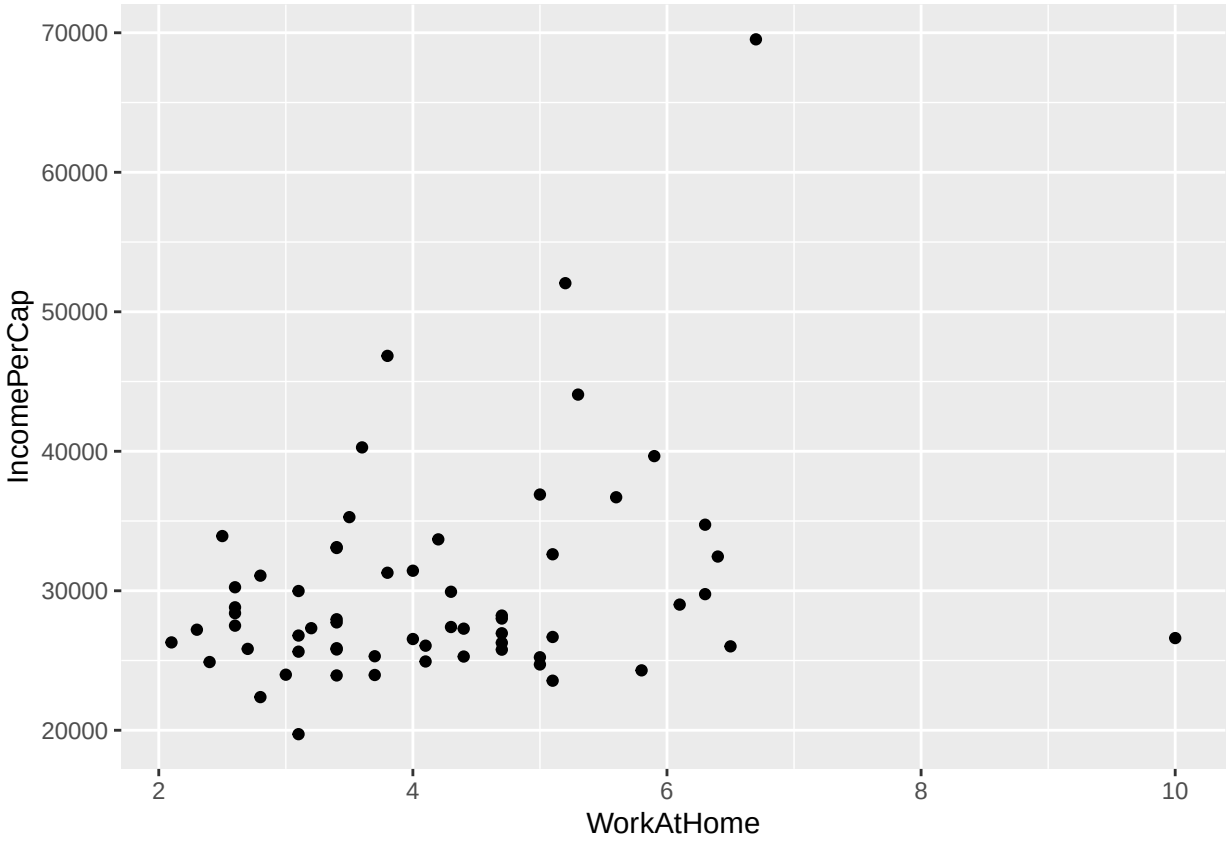
```
cdf %>% ggplot(aes(x=reorder(County, MeanCommute), y=MeanCommute)) + geom_col() + coord_flip()
```



Four of the top 5 NY counties with the longest mean commute times are right here in New York City, Bronx, Queens, Richmond and Kings County.

How has work from home has affected income?

```
cdf %>% ggplot(aes(x=WorkAtHome , y=IncomePerCap)) + geom_point()
```



I am not sure what to make of this scatter plot, it looks like as people work from home loner their income goes up. That might be due to money being saved from commuting.

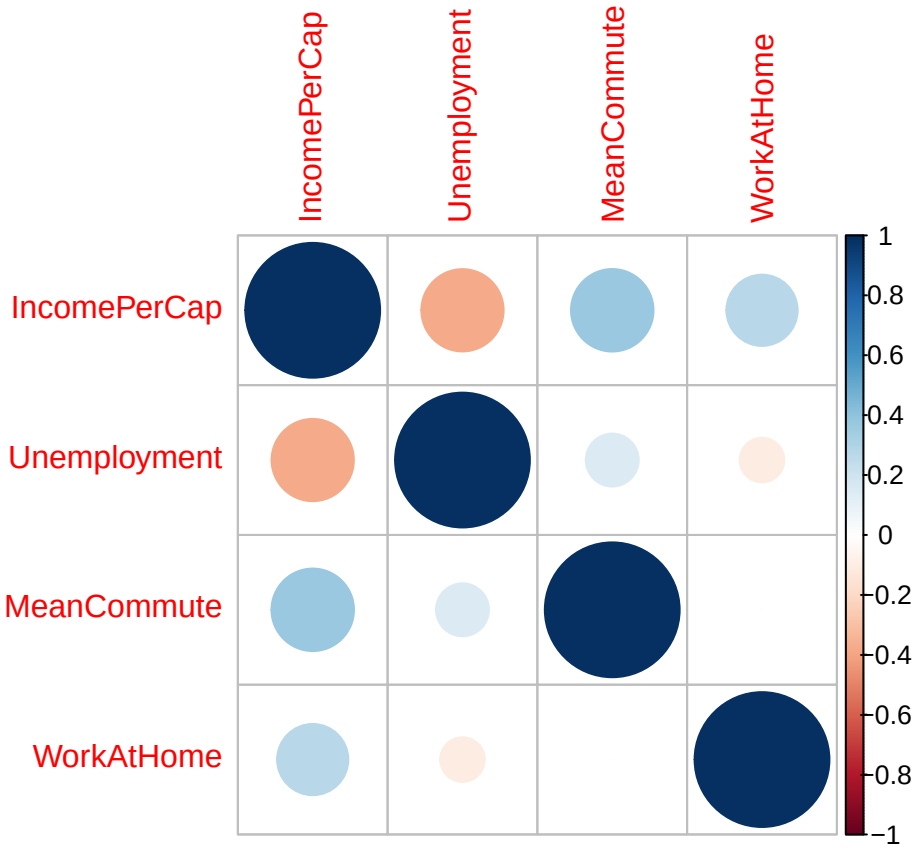
Finally how do these 4 columns relate

```
rownames(cdf) <- cdf$County
cdf %>% select(-c(County))
```

```
##                    IncomePerCap Unemployment MeanCommute WorkAtHome
## Albany County             35278          5.3        20.4        3.5
## Allegany County           22377          7.3        21.4        2.8
## Bronx County              19721         11.6        44.2        3.1
## Broome County             26790          7.2        19.7        3.1
## Cattaraugus County        23984          7.0        21.9        3.0
## Cayuga County             27957          5.9        23.0        3.4
## Chautauqua County         23962          7.7        18.0        3.7
## Chemung County            27209          5.3        19.6        2.3
## Chenango County           25233          6.7        24.1        5.0
## Clinton County            25833          5.8        19.8        2.7
## Columbia County           34737          5.0        25.8        6.3
## Cortland County           26271          4.9        20.5        4.7
## Delaware County           26016          7.1        24.4        6.5
## Dutchess County           36704          6.7        32.2        5.6
## Erie County               31083          5.5        21.3        2.8
## Essex County              29008          6.7        21.9        6.1
## Franklin County           24294          8.2        20.9        5.8
```

```
## Fulton County              26298         7.6        25.0        2.1
## Genesee County             27499         5.1        22.2        2.6
## Greene County              27402         6.2        27.6        4.3
## Hamilton County            24891        11.2        25.3        2.4
## Herkimer County            24932         8.2        23.1        4.1
## Jefferson County           24717         8.0        17.4        5.0
## Kings County               29928         8.1        42.4        4.3
## Lewis County               25779         7.6        24.5        4.7
## Livingston County          25882         4.8        25.5        3.4
## Madison County             28010         4.4        23.3        4.7
## Monroe County              31291         6.6        20.1        3.8
## Montgomery County          25307         7.3        24.5        3.7
## Nassau County              46839         4.9        35.8        3.8
## New York County            69529         6.2        31.8        6.7
## Niagara County             28395         6.2        22.0        2.6
## Oneida County              27283         6.0        19.6        4.4
## Onondaga County            31436         6.4        20.1        4.0
## Ontario County             33685         4.7        23.2        4.2
## Orange County              32616         5.6        33.4        5.1
## Orleans County             23929         6.5        24.1        3.4
## Oswego County              25791         8.6        24.8        3.4
## Otsego County              26688         6.7        22.4        5.1
## Putnam County              44063         5.3        39.2        5.3
## Queens County              28814         6.9        43.5        2.6
## Rensselaer County          33067         5.8        24.4        3.4
## Richmond County            33922         5.5        44.0        2.5
## Rockland County            36898         6.3        31.0        5.0
## St. Lawrence County        23554         8.8        20.9        5.1
## Saratoga County            39653         4.6        25.3        5.9
## Schenectady County         29981         7.3        22.9        3.1
## Schoharie County           26953         7.5        29.7        4.7
## Schuyler County            25285         7.2        24.8        4.4
## Seneca County              26541         4.9        22.3        4.0
## Steuben County             27731         7.0        21.7        3.4
## Suffolk County             40277         5.4        32.0        3.6
## Sullivan County            28224         9.3        30.6        4.7
## Tioga County               30252         6.5        23.0        2.6
## Tompkins County            29759         4.8        18.6        6.3
## Ulster County              32453         7.1        28.2        6.4
## Warren County              33127         4.8        22.8        3.4
## Washington County          26064         6.6        26.2        4.1
## Wayne County               27318         6.4        24.9        3.2
## Westchester County         52049         6.5        34.1        5.2
## Wyoming County             25635         6.5        24.7        3.1
## Yates County               26608         6.0        22.5       10.0
```

```
corrplot(cor(cdf %>% select(-c(County))), method='circle')
```



Foodprices

The foodprices dataset is huge, it takes a full 10 seconds to load on my computer. With 18 columns and 1.8 Million rows there is no surprise on why it takes so long.

```
fdf <- read.csv('/home/kenan/Documents/learning/masters/CUNY-SPS-Masters-DS/DATA_607/projects/project_2/

glimpse(fdf)
```

```
## Rows: 1,829,364
## Columns: 18
## $ adm0_id            <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ adm0_name          <chr> "Afghanistan", "Afghanistan", "Afghanistan", "Afgha~
## $ adm1_id            <int> 272, 272, 272, 272, 272, 272, 272, 272, 272, 272, 2~
## $ adm1_name          <chr> "Badakhshan", "Badakhshan", "Badakhshan", "Badakhsh~
## $ mkt_id             <int> 266, 266, 266, 266, 266, 266, 266, 266, 266, 266, 2~
## $ mkt_name           <chr> "Fayzabad", "Fayzabad", "Fayzabad", "Fayzabad", "Fa~
## $ cm_id              <int> 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55, 55,~
## $ cm_name            <chr> "Bread - Retail", "Bread - Retail", "Bread - Retail~
## $ cur_id             <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ cur_name           <chr> "AFN", "AFN", "AFN", "AFN", "AFN", "AFN", "AFN", "A~
## $ pt_id              <int> 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15, 15,~
## $ pt_name            <chr> "Retail", "Retail", "Retail", "Retail", "Retail", "~
## $ um_id              <int> 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, ~
## $ um_name            <chr> "KG", "KG", "KG", "KG", "KG", "KG", "KG", "KG", "KG~
```

```
## $ mp_month           <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 6, ~
## $ mp_year            <int> 2014, 2014, 2014, 2014, 2014, 2014, 2014, 2014, 201~
## $ mp_price           <dbl> 50.00, 50.00, 50.00, 50.00, 50.00, 50.00, 50.00, 50~
## $ mp_commoditysource <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
```

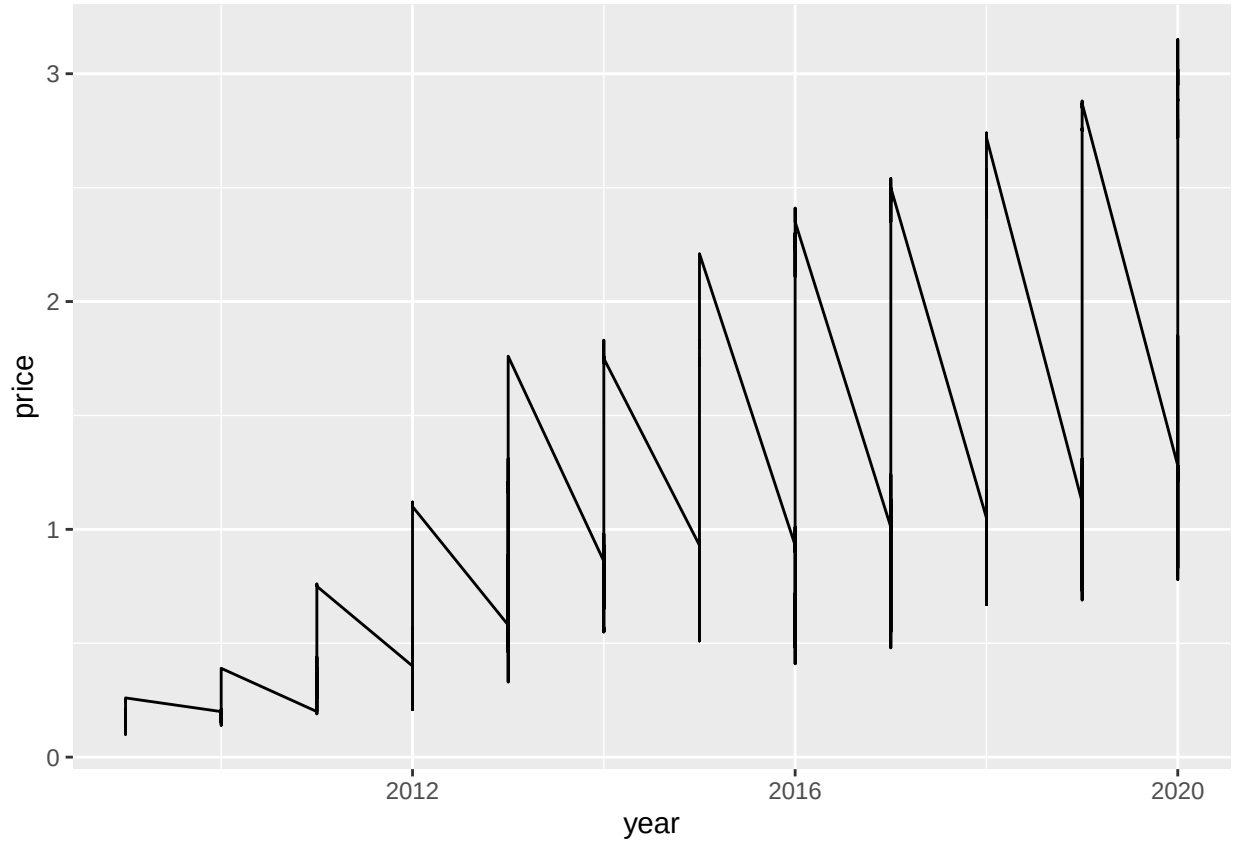
Given the massive size of this dataset, the first thing I'll do is filter a specific region of interest and drop a lot of columns that won't be used. I manage a team from Belarus so that's a good starting point

```
fdf <- fdf %>% filter(adm0_name == 'Belarus') %>% select(c('mp_month', 'mp_year', 'mp_price')) %>% rena

glimpse(fdf)
```

```
## Rows: 429
## Columns: 3
## $ month <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5, 6, 7, 8, 9~
## $ year  <int> 2009, 2009, 2009, 2009, 2009, 2009, 2009, 2009, 2009, 2009, 2009~
## $ price <dbl> 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20, 0.20~
```

How have prices change over the years?

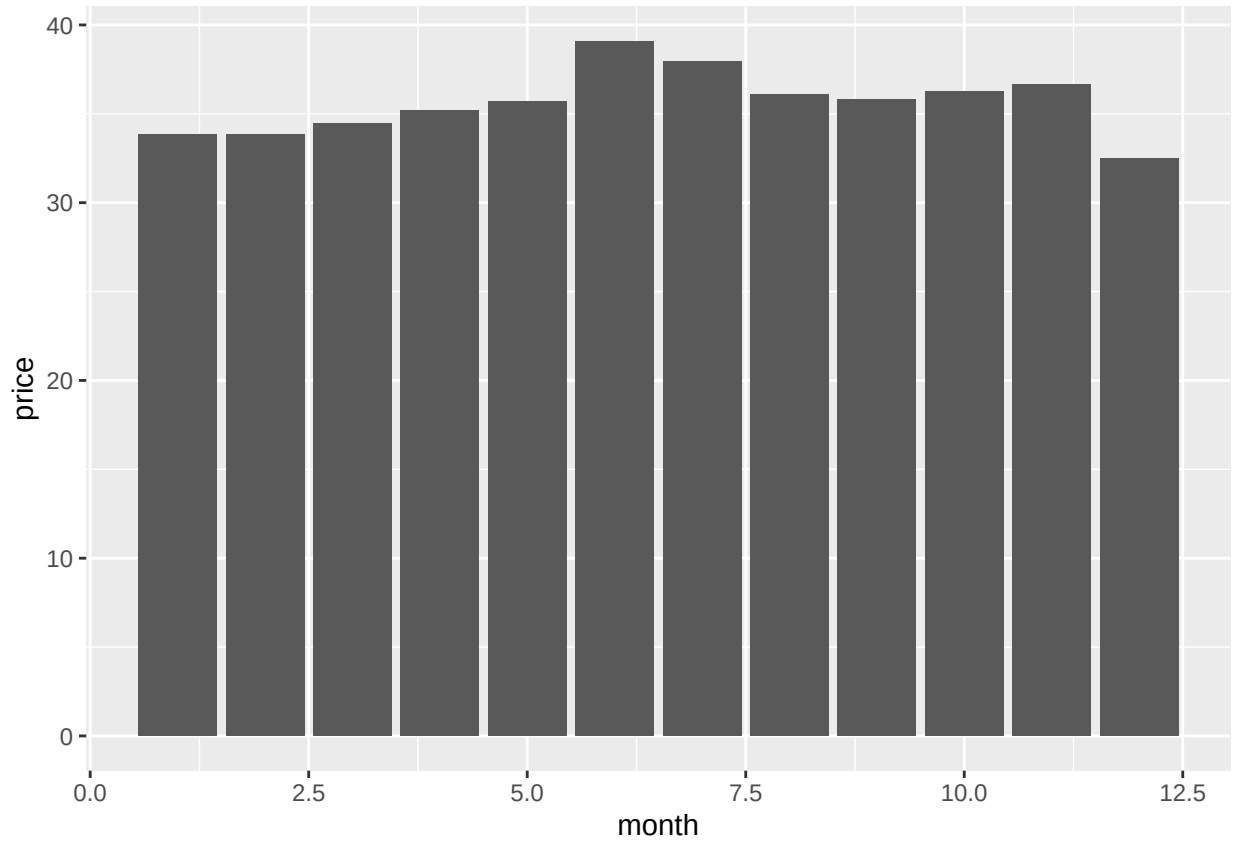
```
fdf %>% ggplot(aes(x=year, y=price)) + geom_line()
```



The prices are trending up but in very strange manner

How prices change on a specific year?

```
fdf %>% ggplot(aes(x=month, y=price)) + geom_col()
```



It looks like prices peak during the middle of the year