DSC 441 HW 1 - Lukasz Grzybek

1.a.

```
#reads the adult.csv file and gives a summary of the data
library(readr)
adult <- read_csv("rr/adult.csv")
```

```
## Rows: 32561 Columns: 15
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (9): workclass, education, marital-status, occupation, relationship, rac...
## dbl (6): age, fnlwgt, education-num, capital-gain, capital-loss, hours-per-week
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
summary(adult)
```

```
##       age         workclass             fnlwgt          education
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character
##  Mean   :38.58                      Mean   : 189778
##  3rd Qu.:48.00                      3rd Qu.: 237051
##  Max.   :90.00                      Max.   :1484705
##  education-num   marital-status      occupation         relationship
##  Min.   : 1.00   Length:32561       Length:32561       Length:32561
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.08
##  3rd Qu.:12.00
##  Max.   :16.00
##      race               sex             capital-gain    capital-loss
##  Length:32561       Length:32561       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1078   Mean   :  87.3
##                                        3rd Qu.:    0   3rd Qu.:   0.0
##                                        Max.   :99999   Max.   :4356.0
##  hours-per-week  native-country     income-bracket
##  Min.   : 1.00   Length:32561       Length:32561
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
```

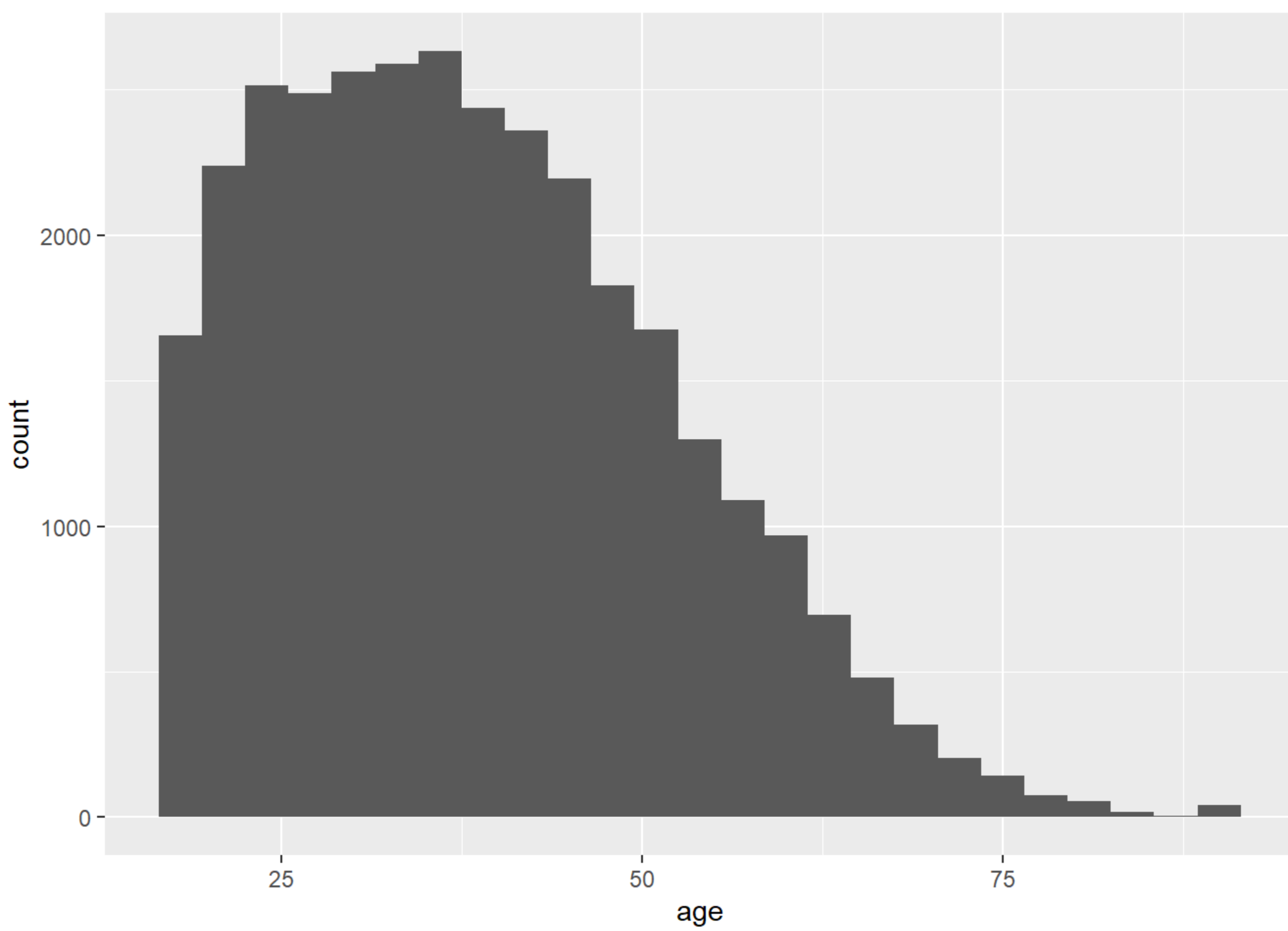
The two chosen variables will be age and education number.

For age, the mean (38.58 years) is slightly larger than the median (37.00 years). This suggests that the distribution is likely at least somewhat positively skewed. The range between the minimum value (17.00 years) and the first quartile (28.00 years) is much smaller than the range between the third quartile (48.00 years) and the maximum value (90.00 years). The median value is also closer to the first quartile as opposed to the third quartile by around 2 years. This further suggests a right skewed (positive) distribution.

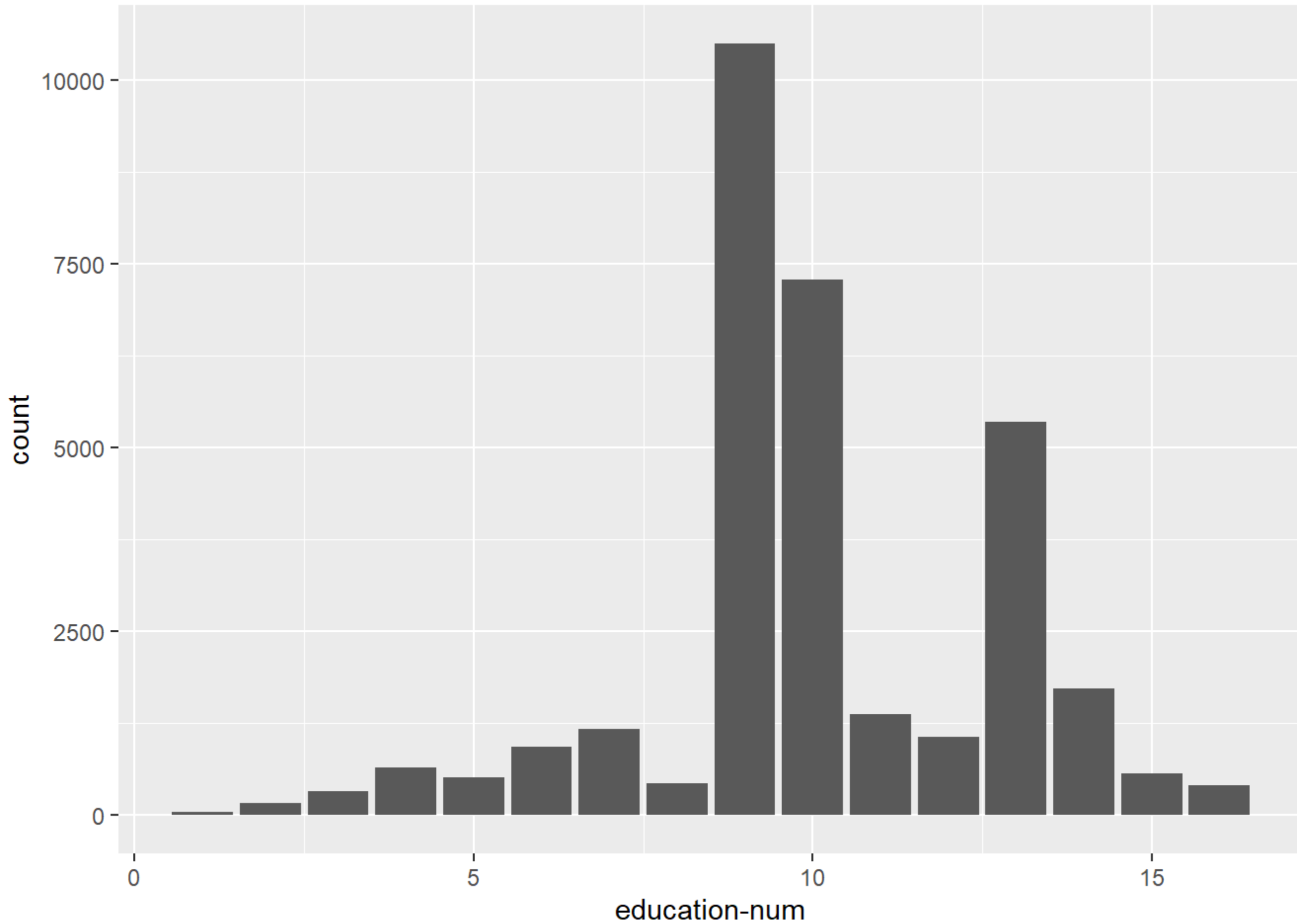
For education number, the median (10) and mean (10.08) are essentially the same as this variable utilizes discrete number with a step value of 1. Therefore, there doesn't seem to be any apparent skew in the data. However, the range between the minimum value (1) and the first quartile (9) is much larger than the range between the third quartile (12) and the maximum value (16). The left whisker being longer than the right one, suggests a negatively skewed distribution. Although, interestingly, the median value is also closer to the first quartile (9) as opposed to the third quartile (12) by around 1 whole point. This is usually seen with positively skewed distributions. It is likely that the minimum value of 1 is an outlier of some sort, and is creating these inconsistent findings in the summary statistics.

1.b.

```
#plots a histogram for age variable
library(ggplot2)
ggplot(adult, aes(age)) + geom_histogram(binwidth = 3)
```



```
##plots a bar graph for age variable
ggplot(adult, aes(`education-num`)) + geom_bar()
```



Upon plotting the ‘age’ variable, a positively skewed normal distribution resulted, which lines up very well with my earlier assumptions.

Upon plotting the ‘education-num’ variable, a bimodal distribution resulted with the mode of the first hump accounting for over 10,000 people, with the second accounting for over 5,000 people. I expected a normal distribution to result (with some few outliers on the lower end of the scale), and with the mean being equal to the median, I expected a more or less bell shaped symmetrical distribution (after accounting for the outliers). However, it turns out that compared to education-numbers 9, 10, and 13, every other category appears much less frequently.

The goal was to compare charts where a certain variable (age or educational-num) was plotted against frequency, and then both charts compared to each other. I chose to use a histogram for age, and a bar graph for educational-num. I couldn’t have both variables be plotted using histograms or bar graphs because ‘age’ was interpreted as a continuous numerical variable, and ‘educational-num’ was interpreted as a categorical variable (since each number represented a level of educational attainment). Furthermore, neither line graphs nor scatter plots would be ideal as they wouldn’t be able to account for a categorical variable, and while histograms don’t either, they at least have the closest resembnlance to bar graphs. A box-plot wasn’t used as various assumptions on how it would roughly look were already made in question 1(a).

1.c.

```
#loads the GGally library in order to plot all numerical variables in a scatterplot matrix bu
t  all categorical variables are first removed from the data for simplicity
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

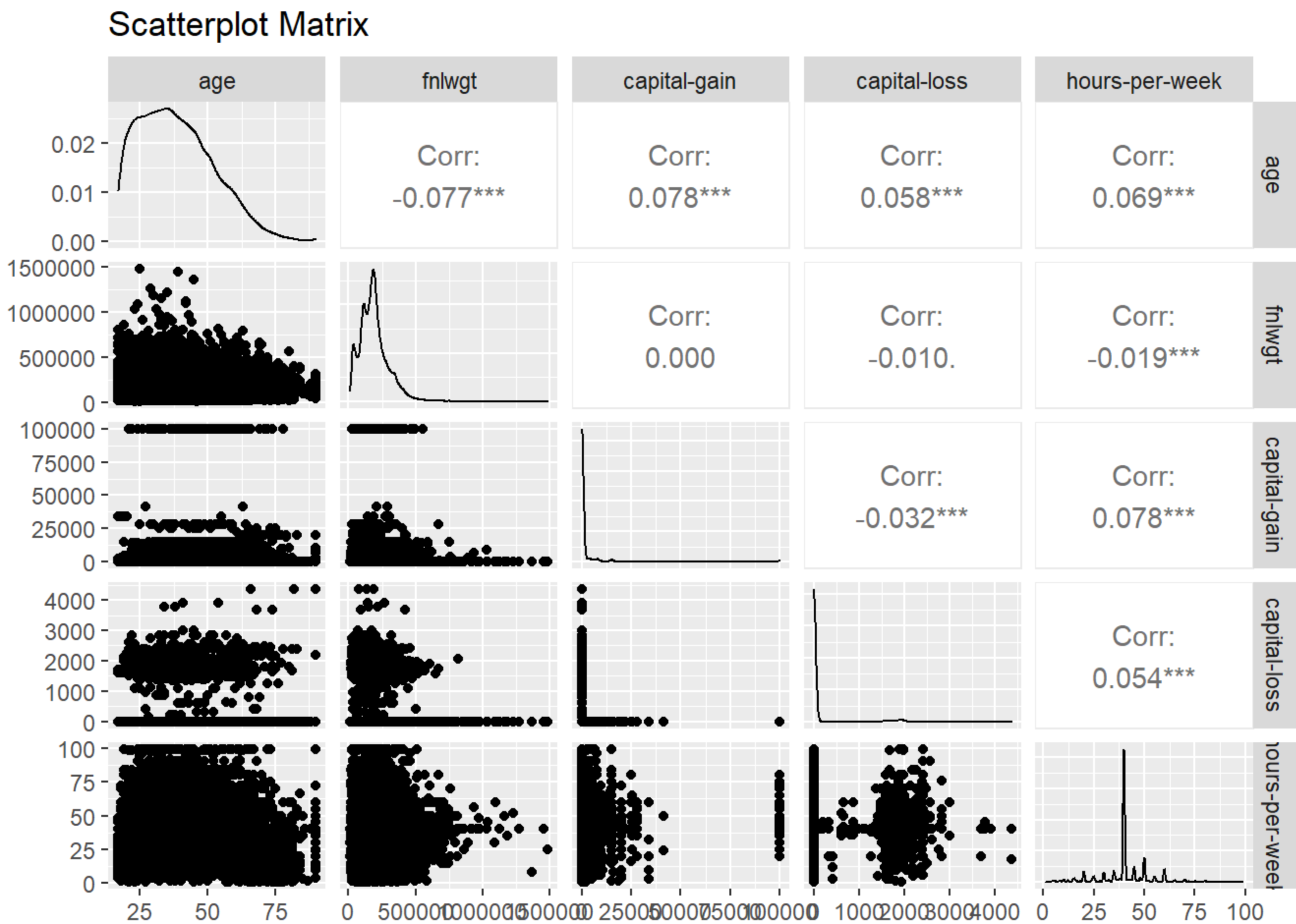
```
library(dplyr)
```

```
##
## 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
```

```
df <- as.data.frame(adult)
df <- df %>% select(-c("workclass", "education", "marital-status", "education-num", "occupati
on", "relationship", "race", "sex", "native-country", "income-bracket"))
ggpairs(df, title="Scatterplot Matrix")
```



This shows how well two numerical variables are correlated to each other, whether they have any correlations, and if they are negative or positive. This would be difficult to determine on distributions.

1.d.

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat <- as.data.frame(adult)
library(dplyr)
dfcat %>%
  group_by(workclass) %>%
  summarise(n = n())
```

workclass <chr>	n <int>
?	1836
Federal-gov	960
Local-gov	2093
Never-worked	7
Private	22696
Self-emp-inc	1116

workclass <chr>	n <int>
Self-emp-not-inc	2541
State-gov	1298
Without-pay	14

9 rows

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(education) %>%
  summarise(n = n())
```

education <chr>	n <int>
10th	933
11th	1175
12th	433
1st-4th	168
5th-6th	333
7th-8th	646
9th	514
Assoc-acdm	1067
Assoc-voc	1382
Bachelors	5355

1-10 of 16 rows Previous **1** 2 Next

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(`marital-status`) %>%
  summarise(n = n())
```

marital-status <chr>	n <int>
Divorced	4443
Married-AF-spouse	23
Married-civ-spouse	14976

marital-status <chr>	n <int>
Married-spouse-absent	418
Never-married	10683
Separated	1025
Widowed	993

7 rows

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(`education-num`) %>%
  summarise(n = n())
```

education-num <dbl>	n <int>
1	51
2	168
3	333
4	646
5	514
6	933
7	1175
8	433
9	10501
10	7291

1-10 of 16 rows Previous **1** 2 Next

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(occupation) %>%
  summarise(n = n())
```

occupation <chr>	n <int>
?	1843
Adm-clerical	3770

occupation <chr>	n <int>
Armed-Forces	9
Craft-repair	4099
Exec-managerial	4066
Farming-fishing	994
Handlers-cleaners	1370
Machine-op-inspct	2002
Other-service	3295
Priv-house-serv	149

1-10 of 15 rows Previous **1** 2 Next

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(relationship) %>%
  summarise(n = n())
```

relationship <chr>	n <int>
Husband	13193
Not-in-family	8305
Other-relative	981
Own-child	5068
Unmarried	3446
Wife	1568

6 rows

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(race) %>%
  summarise(n = n())
```

race <chr>	n <int>
Amer-Indian-Eskimo	311
Asian-Pac-Islander	1039

race <chr>	n <int>
Black	3124
Other	271
White	27816

5 rows

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(sex) %>%
  summarise(n = n())
```

sex <chr>	n <int>
Female	10771
Male	21790

2 rows

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(`native-country`) %>%
  summarise(n = n())
```

native-country <chr>	n <int>
?	583
Cambodia	19
Canada	121
China	75
Columbia	59
Cuba	95
Dominican-Republic	70
Ecuador	28
El-Salvador	106
England	90

1-10 of 42 rows Previous **1** 2 3 4 5 Next

```
#checks the amount of categories in this categorical variable of the adults dataset
dfcat %>%
  group_by(`income-bracket`) %>%
  summarise(n = n())
```

income-bracket <chr>	n <int>
<=50K	24720
>50K	7841

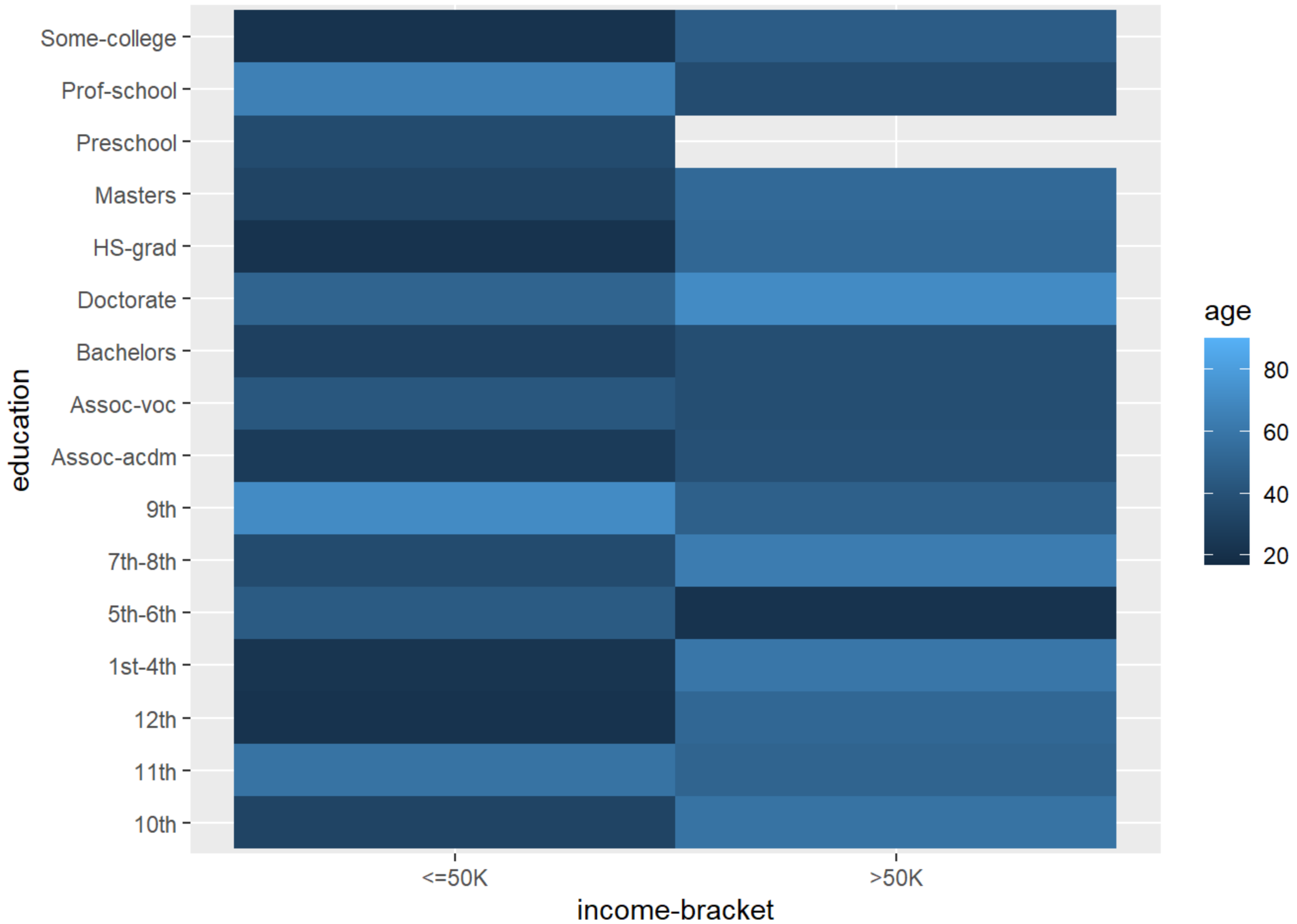
2 rows

1.e.

```
#creates a crosstabulation between education and income-bracket
dfcat %>%
  group_by(education) %>%
  select(education, `income-bracket`) %>%
  table() %>%
  head()
```

```
##          income-bracket
## education <=50K >50K
##   10th      871   62
##   11th     1115   60
##   12th      400   33
##   1st-4th   162    6
##   5th-6th   317   16
##   7th-8th   606   40
```

```
#plots the previous crosstab into a contingency plot
ggplot(dfcat, aes(x=`income-bracket`, y=education, fill=age)) + geom_tile()
```



Based off of the contingency plot, there are more strands of a darker blue in the <=50k column. This makes sense as the darker the strands, the younger the person in question, and it is to be expected that someone just starting in the workforce will earn less than those who have already worked for many years.

2.a. and b.

```
#reads the population_odd and population_even datasets
population_odd <- read_csv("rr/population_odd.csv")
```

```
## Rows: 52 Columns: 7
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): NAME
## dbl (6): STATE, POPESTIMATE2011, POPESTIMATE2013, POPESTIMATE2015, POPESTIMA...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(population_odd)
population_even <- read_csv("rr/population_even.csv")
```

```
## Rows: 52 Columns: 7
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): NAME
## dbl (6): STATE, POPESTIMATE2010, POPESTIMATE2012, POPESTIMATE2014, POPESTIMA...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(population_even)
```

```
#joins population_even and population_odd, removes the excess STATE column, organizes the col
umns by year, and renames each column to just the year
newtab = population_odd %>% inner_join(population_even, by="NAME") %>%
  select(-c("STATE.y")) %>%
  relocate(POPESTIMATE2010, .before = POPESTIMATE2011) %>%
  relocate(POPESTIMATE2012, .before = POPESTIMATE2013) %>%
  relocate(POPESTIMATE2014, .before = POPESTIMATE2015) %>%
  relocate(POPESTIMATE2016, .before = POPESTIMATE2017) %>%
  relocate(POPESTIMATE2018, .before = POPESTIMATE2019) %>%
  rename(STATE = STATE.x) %>%
  rename('2010' = POPESTIMATE2010) %>%
  rename('2011' = POPESTIMATE2011) %>%
  rename('2012' = POPESTIMATE2012) %>%
  rename('2013' = POPESTIMATE2013) %>%
  rename('2014' = POPESTIMATE2014) %>%
  rename('2015' = POPESTIMATE2015) %>%
  rename('2016' = POPESTIMATE2016) %>%
  rename('2017' = POPESTIMATE2017) %>%
  rename('2018' = POPESTIMATE2018) %>%
  rename('2019' = POPESTIMATE2019)

head(newtab)
```

ST... <dbl>	NAME <chr>	2010 <dbl>	2011 <dbl>	2012 <dbl>	2013 <dbl>	2014 <dbl>	2015 <dbl>	2016 <dbl>
1	Alabama	4785437	4799069	4815588	4830081	4841799	4852347	4863525
2	Alaska	713910	722128	730443	737068	736283	737498	741456
4	Arizona	6407172	NA	6554978	6632764	6730413	6829676	6941072
5	Arkansas	2921964	2940667	2952164	2959400	2967392	2978048	2989918
6	California	37319502	37638369	37948800	38260787	38596972	38918045	39167117
8	Colorado	5047349	5121108	5192647	5269035	5350101	5450623	5539215

6 rows | 1-9 of 12 columns

2.c.

```
#finds summary stats of each year
summary(newtab$'2010')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   564487  1764843  4092836  6020061  6610438 37319502
```

```
summary(newtab$'2011')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##   567299  1712291  3872036  6054176  6720105 37638369        1
```

```
summary(newtab$'2012')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   576305  1788808  4142674  6105105  6721518 37948800
```

```
summary(newtab$'2013')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##   582122  1732560  3922468  6039414  6673040 38260787        1
```

```
summary(newtab$'2014')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   582531  1794895  4188796  6189152  6835611 38596972
```

```
summary(newtab$'2015')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##   585613  1866664  4425976  6322693  6996666 38918045        1
```

```
summary(newtab$'2016')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   584215  1793862  4264079  6275923  7029497 39167117
```

```
summary(newtab$'2017')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##   578931  1866476  4452268  6416830  7233685 39358497        1
```

```
summary(newtab$'2018')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   577601  1790852  4321520  6343863  7249485 39461588
```

```
summary(newtab$'2019')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##   578759  1789606  4217737  6384525  7446805 39512223        1
```

Missing NA's found in the years: 2011, 2013, 2015, 2017, 2019.

The missing values will be replaced by the average of the surrounding years.

((mean 2010) + (mean 2012))/2 = 2011's NA (6020061 + 6105105)/2 = 6062583

((mean 2012) + (mean 2014))/2 = 2013's NA (6105105 + 6189152)/2 = 6147128.5

((mean 2014) + (mean 2016))/2 = 2015's NA (6189152 + 6275923)/2 = 6232537.5

((mean 2016) + (mean 2018))/2 = 2017's NA (6275923 + 6343863)/2 = 6309893

(mean 2018) = 2019's NA 6343863

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ tibble  3.1.8     ✓ stringr 1.4.1
## ✓ tidyr   1.2.1     ✓ forcats 0.5.2
## ✓ purrr   0.3.4     
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
```

```
#missing values replaced by the average of the surrounding years.
newtab$'2011' <- newtab$'2011' %>%
    replace_na(6062583)
summary(newtab$'2011')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   567299  1776482  4120928  6054337  6666844 37638369
```

```
newtab$'2013' <- newtab$'2013' %>%
    replace_na(6147128.5)
summary(newtab$'2013')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   582122  1793237  4163564  6041485  6652902 38260787
```

```
newtab$'2015' <- newtab$'2015' %>%
    replace_na(6232537.5)
summary(newtab$'2015')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   585613  1878970  4545302  6320959  6913171 38918045
```

```
newtab$'2017' <- newtab$'2017' %>%
    replace_na(6309893)
summary(newtab$'2017')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   578931  1891211  4561414  6414774  7138846 39358497
```

```
newtab$'2019' <- newtab$'2019' %>%
    replace_na(6343863)
summary(newtab$'2019')
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##   578759  1790876  4342705  6383743  7362761 39512223
```

All missing NA's have been replaced.

2.d.a.

```
#max value column created for every row of the dataset
newtab %>%
  rowwise() %>%
  mutate(MAX = max(c(`2010`, `2011`, `2012`, `2013`, `2014`, `2015`, `2016`, `2017`, `2018`,
`2019`))) %>%

  head()
```

ST... <dbl>	NAME <chr>	2010 <dbl>	2011 <dbl>	2012 <dbl>	2013 <dbl>	2014 <dbl>	2015 <dbl>	2016 <dbl>
1	Alabama	4785437	4799069	4815588	4830081	4841799	4852347	4863525
2	Alaska	713910	722128	730443	737068	736283	737498	741456

ST… <dbl>	NAME <chr>	2010 <dbl>	2011 <dbl>	2012 <dbl>	2013 <dbl>	2014 <dbl>	2015 <dbl>	2016 <dbl>
4	Arizona	6407172	6062583	6554978	6632764	6730413	6829676	6941072
5	Arkansas	2921964	2940667	2952164	2959400	2967392	2978048	2989918
6	California	37319502	37638369	37948800	38260787	38596972	38918045	39167117
8	Colorado	5047349	5121108	5192647	5269035	5350101	5450623	5539215

6 rows | 1-9 of 13 columns

2.d.b.

```
#total column created for every row
newtab %>%
  rowwise() %>%
  mutate(TOTAL = sum(c(`2010`, `2011`, `2012`, `2013`, `2014`, `2015`, `2016`, `2017`, `2018
`, `2019`))) %>%

  head()
```

ST… <dbl>	NAME <chr>	2010 <dbl>	2011 <dbl>	2012 <dbl>	2013 <dbl>	2014 <dbl>	2015 <dbl>	2016 <dbl>
1	Alabama	4785437	4799069	4815588	4830081	4841799	4852347	4863525
2	Alaska	713910	722128	730443	737068	736283	737498	741456
4	Arizona	6407172	6062583	6554978	6632764	6730413	6829676	6941072
5	Arkansas	2921964	2940667	2952164	2959400	2967392	2978048	2989918
6	California	37319502	37638369	37948800	38260787	38596972	38918045	39167117
8	Colorado	5047349	5121108	5192647	5269035	5350101	5450623	5539215

6 rows | 1-9 of 13 columns

Getting the total (when the max was already calculated), only required a minor change in the code because the only thing we needed to switch out was the type of function we were using for mutating into a new column.

2.e.

```
#totals for each year found
sum(newtab$`2010`)
```

```
## [1] 313043191
```

```
sum(newtab$`2011`)
```

```
## [1] 314825546
```

```
sum(newtab$`2012`)
```

```
## [1] 317465478
```

```
sum(newtab$`2013`)
```

```
## [1] 314157237
```

```
sum(newtab$`2014`)
```

```
## [1] 321835882
```

```
sum(newtab$`2015`)
```

```
## [1] 328689874
```

```
sum(newtab$`2016`)
```

```
## [1] 326347983
```

```
sum(newtab$`2017`)
```

```
## [1] 333568236
```

```
sum(newtab$`2018`)
```

```
## [1] 329880855
```

```
sum(newtab$`2019`)
```

```
## [1] 331954646
```

3.

```
#population_odd and population_even joined, extra STATE column removed, and everything organi
zed by year again, state column renamed
newtab3 = population_odd %>% inner_join(population_even, by="NAME") %>%
  select(-c("STATE.y")) %>%
  relocate(POPESTIMATE2010, .before = POPESTIMATE2011) %>%
  relocate(POPESTIMATE2012, .before = POPESTIMATE2013) %>%
  relocate(POPESTIMATE2014, .before = POPESTIMATE2015) %>%
  relocate(POPESTIMATE2016, .before = POPESTIMATE2017) %>%
  relocate(POPESTIMATE2018, .before = POPESTIMATE2019) %>%
  rename(STATE = STATE.x)


head(newtab3)
```

ST... <dbl>	NAME <chr>	POPESTIMATE2... <dbl>	POPESTIMATE2... <dbl>	POPESTIMATE2... <dbl>	POPESTIMATE2... <dbl>
1	Alabama	4785437	4799069	4815588	4830081
2	Alaska	713910	722128	730443	737068
4	Arizona	6407172	NA	6554978	6632764
5	Arkansas	2921964	2940667	2952164	2959400
6	California	37319502	37638369	37948800	38260787
8	Colorado	5047349	5121108	5192647	5269035

6 rows | 1-6 of 12 columns

```
#data reshaped with the years now belonging to a single column and their values belonging und
er a single column
newtab3 <- newtab3 %>%
  rownames_to_column(var = "STATES") %>%
  pivot_longer(cols = c("POPESTIMATE2010", "POPESTIMATE2011","POPESTIMATE2012", "POPESTIMATE2
013","POPESTIMATE2014", "POPESTIMATE2015","POPESTIMATE2016", "POPESTIMATE2017", "POPESTIMATE2
018", "POPESTIMATE2019"), names_to = "Year", values_to = "Population")
```

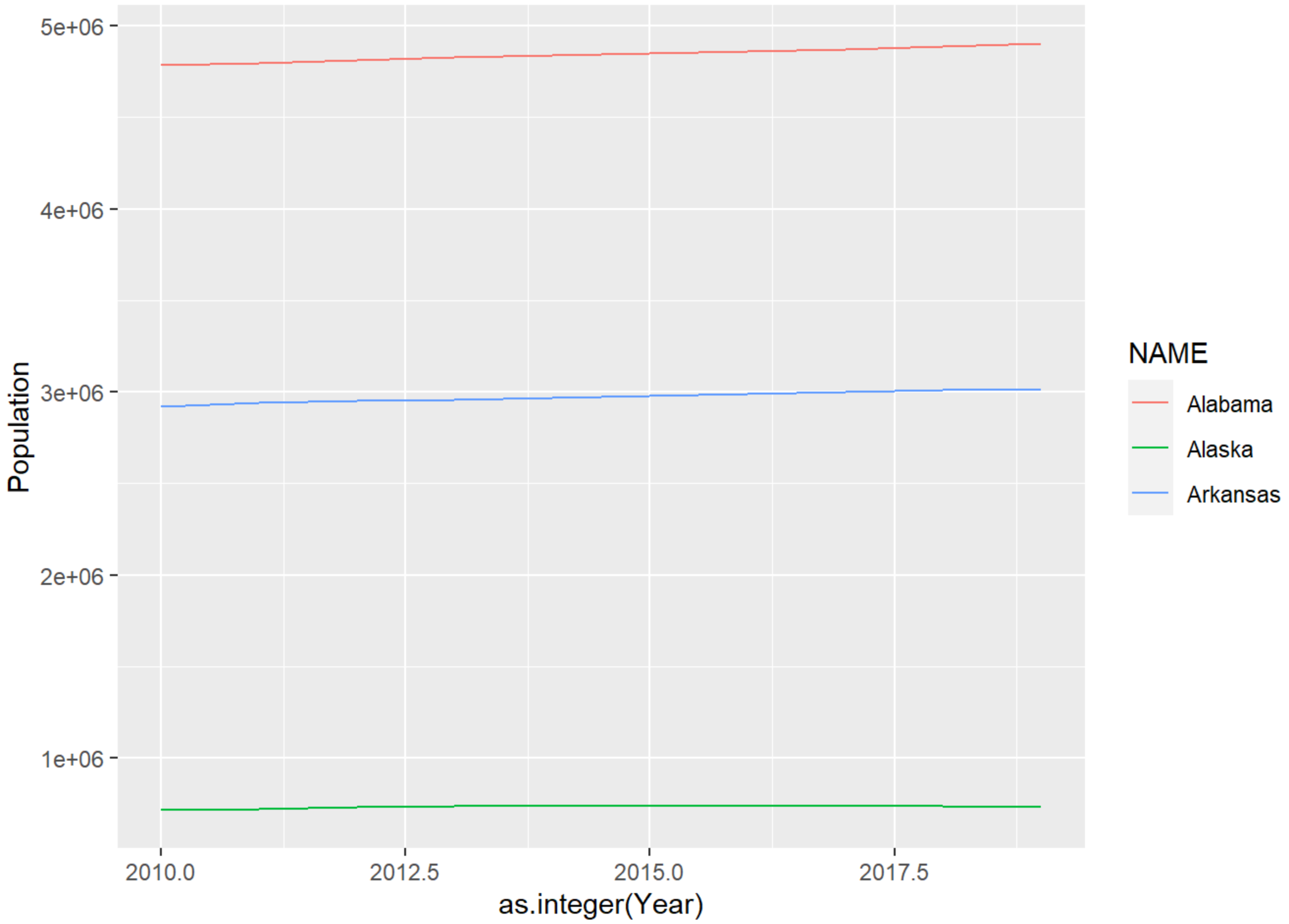
```
#rows of "Years" column renamed to just the numerical string
newtab3$Year[newtab3$Year=="POPESTIMATE2010"] <- "`2010`"
newtab3$Year[newtab3$Year=="`2010`"] <- "2010"
newtab3$Year[newtab3$Year=="POPESTIMATE2011"] <- "2011"
newtab3$Year[newtab3$Year=="POPESTIMATE2012"] <- "2012"
newtab3$Year[newtab3$Year=="POPESTIMATE2013"] <- "2013"
newtab3$Year[newtab3$Year=="POPESTIMATE2014"] <- "2014"
newtab3$Year[newtab3$Year=="POPESTIMATE2015"] <- "2015"
newtab3$Year[newtab3$Year=="POPESTIMATE2016"] <- "2016"
newtab3$Year[newtab3$Year=="POPESTIMATE2017"] <- "2017"
newtab3$Year[newtab3$Year=="POPESTIMATE2018"] <- "2018"
newtab3$Year[newtab3$Year=="POPESTIMATE2019"] <- "2019"
head(newtab3)
```

STATES <chr>	STATE <dbl>	NAME <chr>	Year <chr>	Population <dbl>
1	1	Alabama	2010	4785437
1	1	Alabama	2011	4799069
1	1	Alabama	2012	4815588
1	1	Alabama	2013	4830081
1	1	Alabama	2014	4841799
1	1	Alabama	2015	4852347

6 rows

```
#3 states chosen, the other rows belonging to other states are removwd
newtab3 <- newtab3[-c(41:520), ]
newtab3 <- newtab3[-c(21:30), ]
```

```
#data turned into data frame and then plotted on a line graph
dfnewtab3 <- as.data.frame(newtab3)
plt <- ggplot(dfnewtab3, aes(x=as.integer(Year), y=Population, color=NAME))
plt + geom_line()
```




```
head(newtab3)
```

STATES <chr>	STATE <dbl>	NAME <chr>	Year <chr>	Population <dbl>
1	1	Alabama	2010	4785437
1	1	Alabama	2011	4799069
1	1	Alabama	2012	4815588
1	1	Alabama	2013	4830081
1	1	Alabama	2014	4841799
1	1	Alabama	2015	4852347

6 rows

4.A. One way data can be dirty is when it is inconsistent due to being taken from multiple sources, with each source utilizing a different scale. One solution to this would be to normalize the data to a shared scale.

Another way data can be dirty is when it is incomplete due to missing values during human entry. A possible solution to this would be to simply remove the rows of data with the missing values (assuming that it is a large enough dataset and that it would be appropriate).

B.

a. I would use clustering to help with figuring out the five groups of customers who buy similar things, where each row is a customer and there are columns that describe their purchases. Clustering this data would help establish these groups from scratch based on some similarity (in this case the customer's purchases).

b. Using classification and predication, one can predict if a customer will buy milk based on what else they bought. This is because there are already pre-established groups that are centered around how customers make their purchases. There will likely be a group of people who buy milk along with certain other items, and another group that don't buy milk with those other items. If a new customer buys one of those other items and it happens to be one that others in his position bought along with milk, then I can assume that the new customer will likely purchase milk as well.

c.Using association rule mining, which looks at two events occurring together, it can be used to check and determine what different sets of products are often purchased together.

C.

a. Organizing the customers of a company according to education level. - It is not a data mining task because you are just sorting data based on some attribute. b.Computing the total sales of a company - It is not a data mining task because you are just performming a simple calculation from a dataset. c.Sorting a student database according to identification numbers - - It is not a data mining task because you are also just sorting data based on some attribute. d.Predicting the outcomes of tossing a (fair) pair of dice - It is not data mining task because it has more to do with calculating probabilty. e.Predicting the future stock price of a company using historical records - It is a data mining task because classification and prediction can be used here, specifically numerical regression.