HW1

sampras

01/29/2025

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(ggplot2)      # For visualizations
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
library(dplyr)        # For data manipulation
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## 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(tidyr)        # For reshaping data
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```
library(readr)        # To read csv files
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```
setwd("C:/Users/SAM/OneDrive/Documents/data science/as 1")

adult <- read.csv("adult.csv")

str(adult)
```

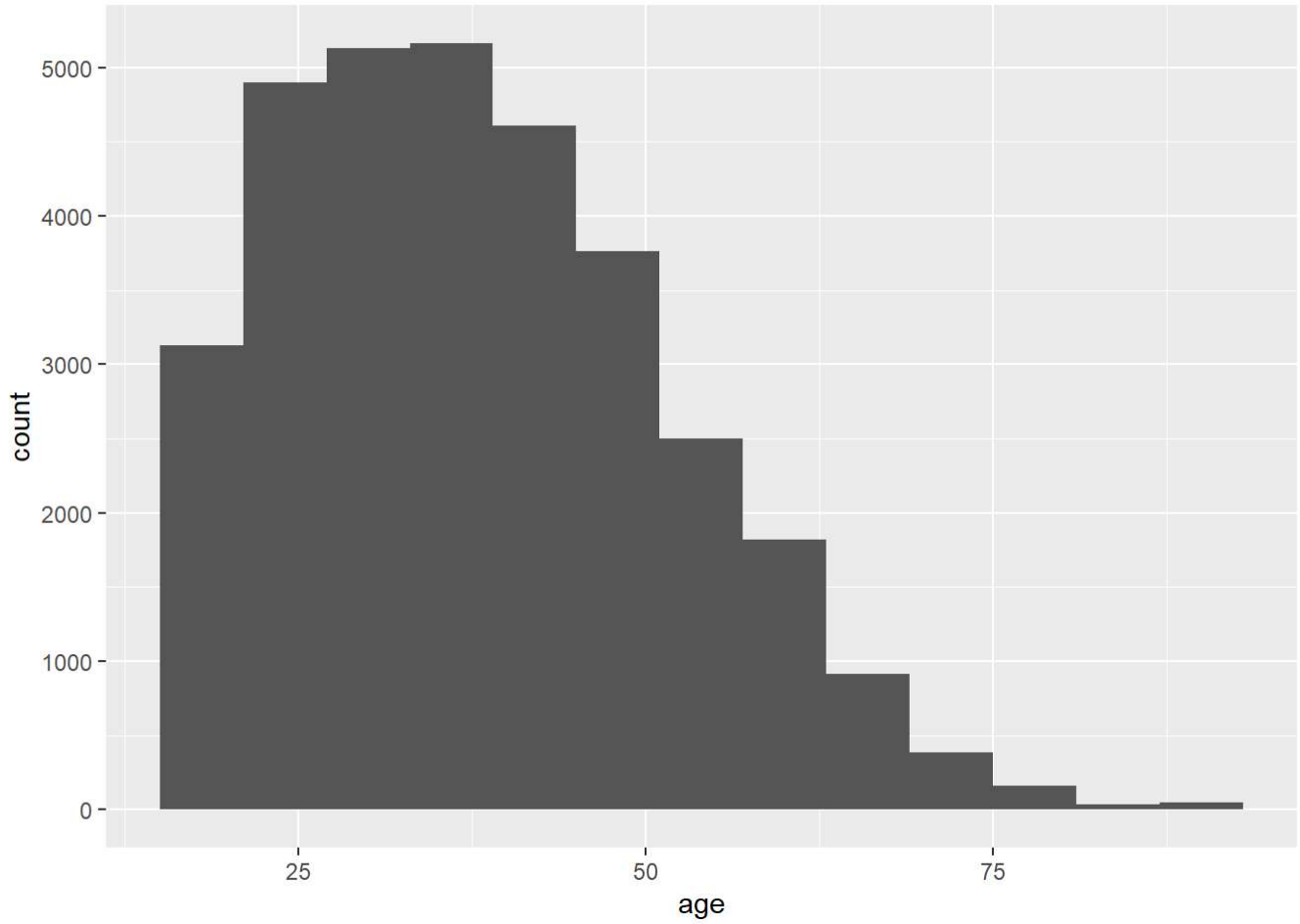
```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass     : chr  " State-gov" " Self-emp-not-inc" " Private" " Private" ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449
...
##  $ education     : chr  " Bachelors" " Bachelors" " HS-grad" " 11th" ...
##  $ education.num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status: chr  " Never-married" " Married-civ-spouse" " Divorced" " Married-civ-spou
se" ...
##  $ occupation    : chr  " Adm-clerical" " Exec-managerial" " Handlers-cleaners" " Handlers-cl
eaners" ...
##  $ relationship  : chr  " Not-in-family" " Husband" " Not-in-family" " Husband" ...
##  $ race          : chr  " White" " White" " White" " Black" ...
##  $ sex           : chr  " Male" " Male" " Male" " Male" ...
##  $ capital.gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ capital.loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country: chr  " United-States" " United-States" " United-States" " United-States"
...
##  $ income.bracket: chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

```
# 1(a)
summary(adult)
```

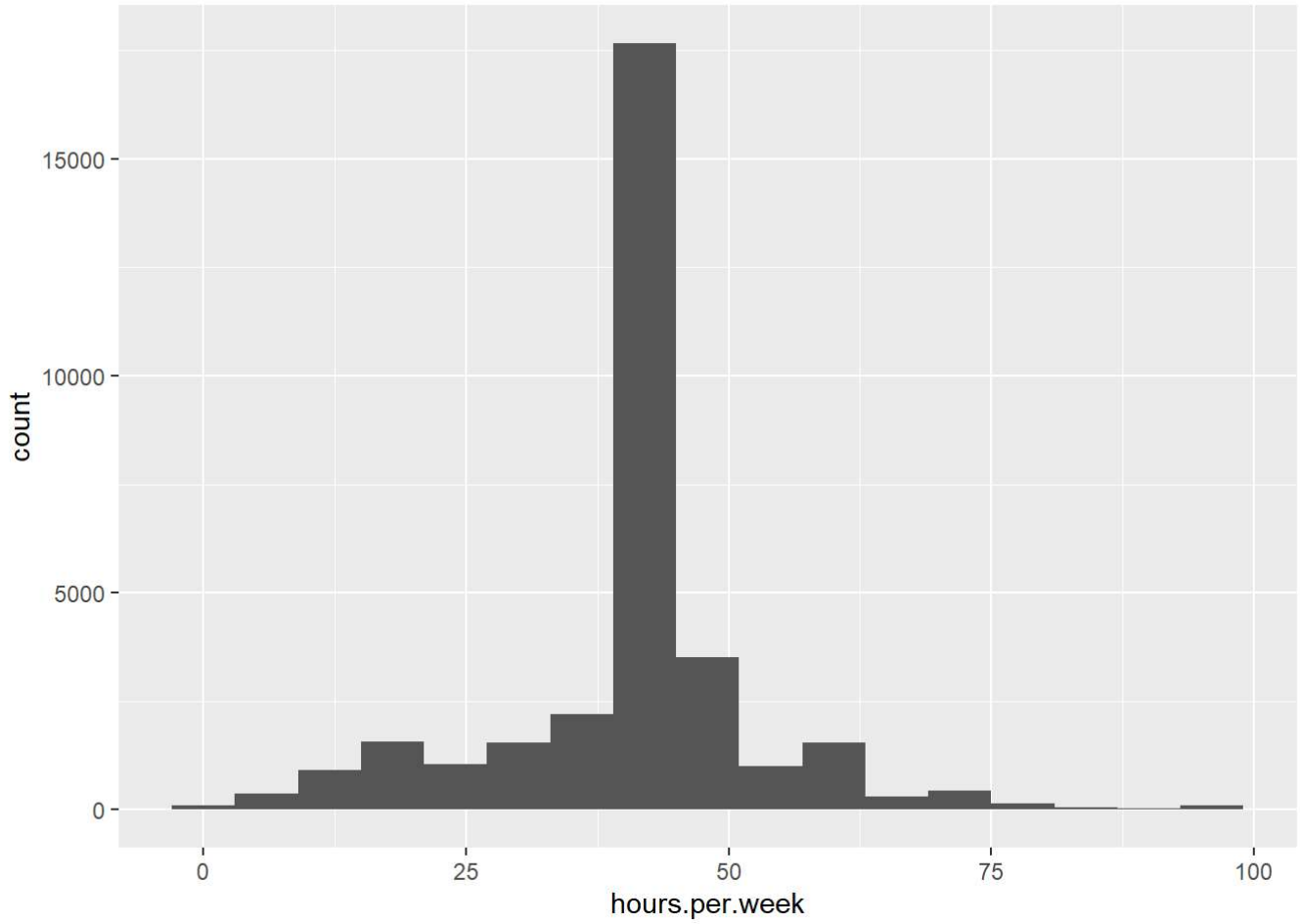
```
##       age         workclass             fnlwgt         education
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character
##  Mean   :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
```

```
#Based on the Summary of the Two Variables: Age and Hours per Week
#Both age and hours worked per week have mean values that closely align with their medians, sugg
esting nearly symmetric distributions.
#Age exhibits a wider range, with most individuals between 28 and 48 years old, whereas hours pe
r week is more concentrated, with the majority working between 40 and 45 hours.
#The maximum age of 90 indicates slight right skewness due to older individuals, while the maxim
um of 99 hours per week suggests minor right skewness, likely from outliers.
#Both variables have relatively small interquartile ranges, indicating that most of the populati
on falls within a narrow range of values.
```

```
# 1(b)
# Visualisation for Age

ggplot(adult, aes(age))+geom_histogram(binwidth = 06)
```

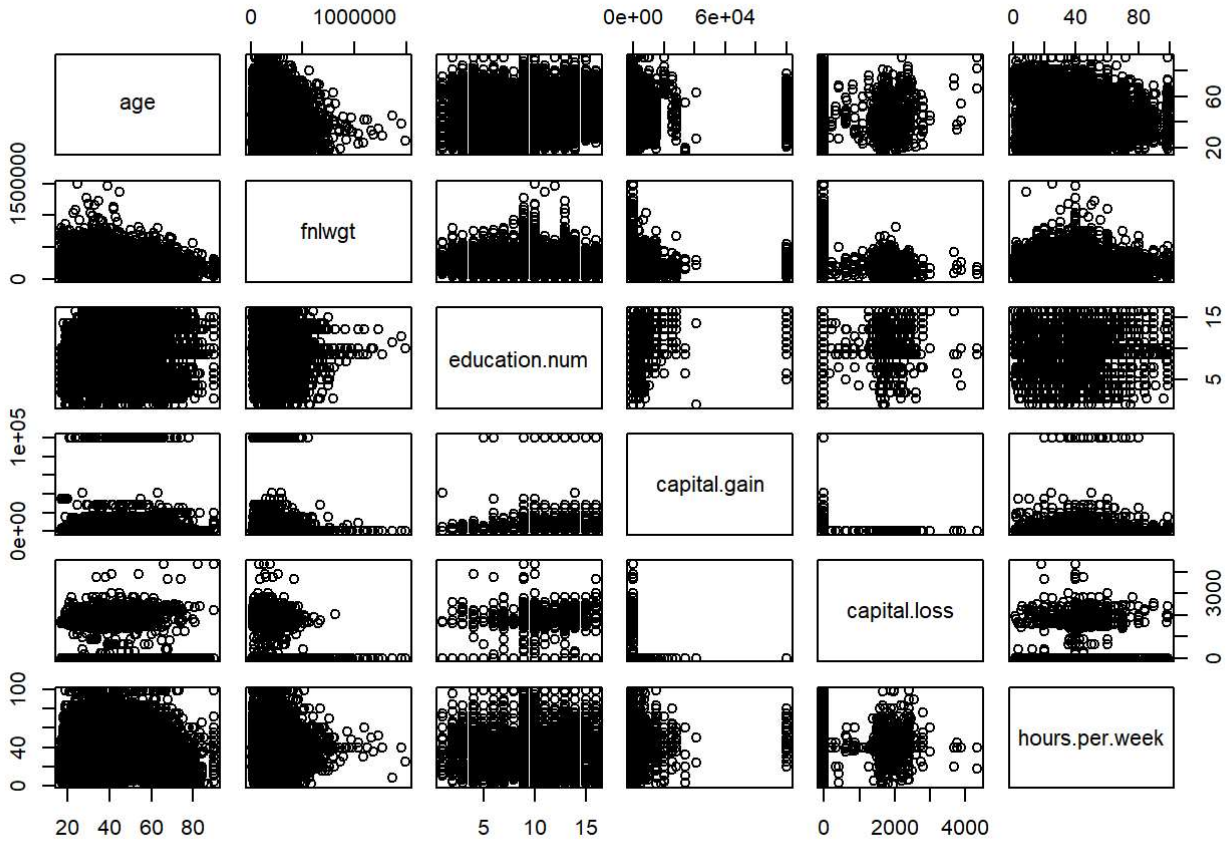



```
# Visualization for Hours per Week

ggplot(adult, aes(hours.per.week))+geom_histogram(binwidth = 06)
```




```
#I selected a histogram for my presentation because it is straightforward and easy to interpret. Histograms group data into bins, making patterns easy to spot.
#The histogram for "age" reveals a tail extending towards older ages, with a larger concentration of observations in younger age groups, which aligns with the slightly right-skewed distribution described in the summary statistics.
# The distribution of hours worked per week is predominantly centered around 40 hours, reflecting that most people work full-time. There's a slight right skew, with a few individuals working considerably more than 40 hours, including some outliers reaching up to 100 hours per week.
# Overall, the visualizations corroborate the hypotheses from part (a) based on the summary statistics.
```

```
# 1(c)

numerical_adult <- adult[, sapply(adult, is.numeric)]
pairs(numerical_adult)
```



#The scatterplot matrix provides valuable insights into the relationships between age and hours worked per week.
#Age vs. fnlwgt, education.num, and hours worked per week: There is no evident strong linear relationship between age and these variables. However, the age vs. education.num plot suggests a slight upward trend, indicating that older individuals tend to have somewhat higher levels of education.
#Hours worked per week: This variable appears mostly independent of the others, as there is no noticeable correlation in the pairwise plots.
#Overall, while no strong linear relationships are found, some weak associations exist, particularly between age and education.num. The scatterplot matrix effectively shows the distribution and clustering of the data, but most variables do not reveal significant interactions.

```
# 1(d)
# Considering the data from the workclass and group_by education

library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.3.3
```

```
## Warning: package 'tibble' was built under R version 4.3.3
```

```
## Warning: package 'purrr' was built under R version 4.3.3
```

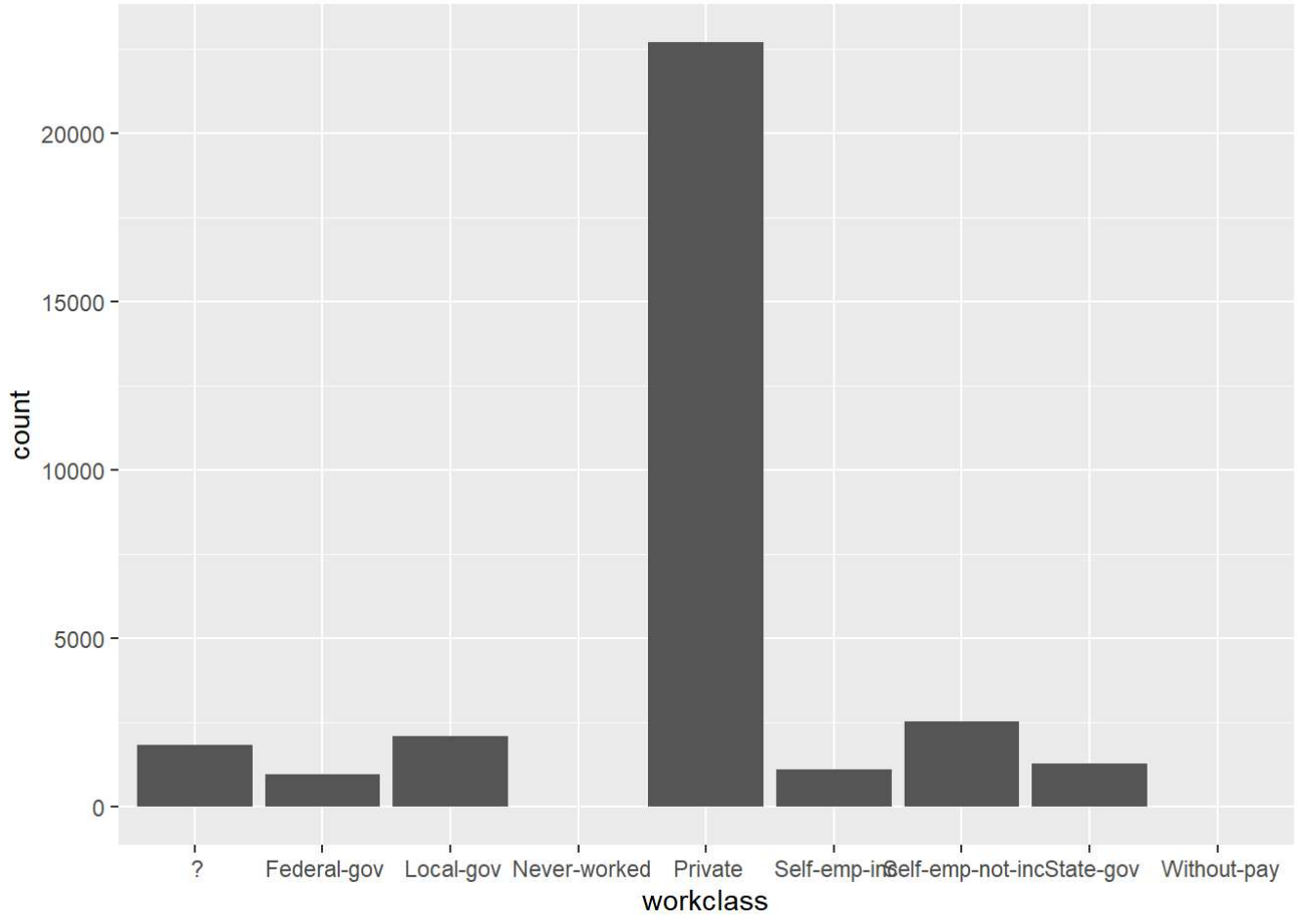
```
## Warning: package 'stringr' was built under R version 4.3.3
```

```
## Warning: package 'forcats' was built under R version 4.3.3
```

```
## Warning: package 'lubridate' was built under R version 4.3.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.4     ✓ tibble    3.2.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
ggplot(adult, aes(x = workclass))+geom_bar()
```



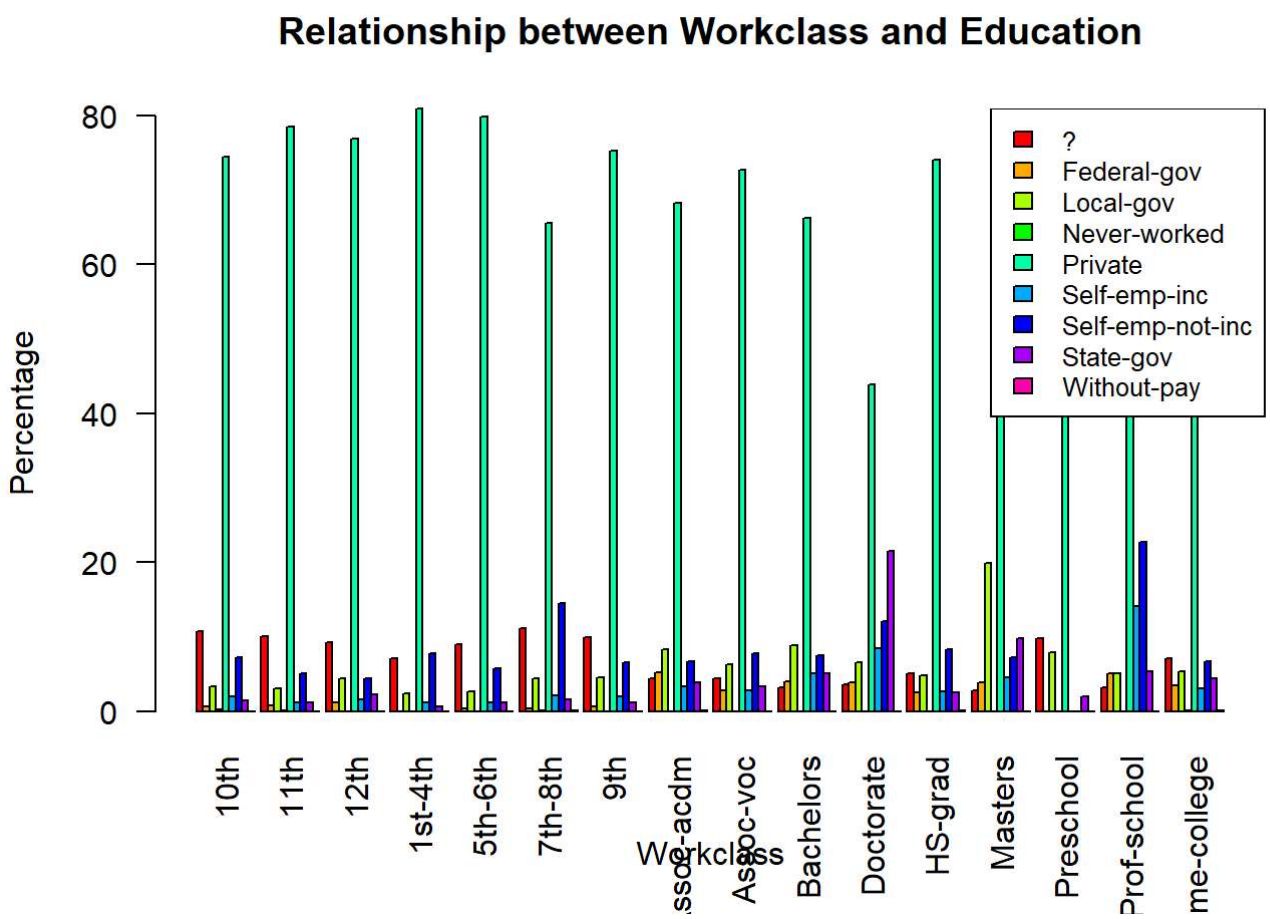
```
education_counts <- adult %>%
 group_by(education) %>%
 count() %>%
 arrange(desc(n))

education_counts
```

```
## # A tibble: 16 × 2
## # Groups:   education [16]
##    education           n
##    <chr>           <int>
##  1 " HS-grad"      10501
##  2 " Some-college"  7291
##  3 " Bachelors"     5355
##  4 " Masters"       1723
##  5 " Assoc-voc"     1382
##  6 " 11th"          1175
##  7 " Assoc-acdm"    1067
##  8 " 10th"           933
##  9 " 7th-8th"        646
## 10 " Prof-school"    576
## 11 " 9th"            514
## 12 " 12th"           433
## 13 " Doctorate"      413
## 14 " 5th-6th"        333
## 15 " 1st-4th"        168
## 16 " Preschool"       51
```

```
# 1(e)

#Our goal here is to explore the relationship between the type of work a person does and their l
evel of education. The data reveals some intriguing trends.

# For instance, in the "State government" sector, there appears to be a higher concentration of
individuals with mid-level education, such as high school diplomas or some college experience. I
n contrast, the "Federal government" sector shows a clear tendency toward a more highly educated
workforce, with many individuals holding bachelor's degrees or higher. This indicates that feder
al positions likely have stricter educational requirements than state-level roles, which seem to
attract a broader range of educational backgrounds.
# These observations imply that the type of work a person engages in could be associated with th
eir level of education.

# Create a contingency table
cross_table <- table(adult$workclass, adult$education)
print(cross_table)
```

```
##
##                     10th  11th  12th  1st-4th  5th-6th  7th-8th  9th
##    ?                 100   118    40       12       30       72   51
##    Federal-gov         6     9     5        0        1        2    3
##    Local-gov          31    36    19        4        9       28   23
##    Never-worked        2     1     0        0        0        1    0
##    Private           695   923   333      136      266      424  387
##    Self-emp-inc       19    14     7        2        4       14   10
##    Self-emp-not-inc   67    60    19       13       19       94   34
##    State-gov          13    14    10        1        4       10    6
##    Without-pay         0     0     0        0        0        1    0
##
##                     Assoc-acdm  Assoc-voc  Bachelors  Doctorate  HS-grad
##    ?                        47         61        173         15      532
##    Federal-gov              55         38        212         16      263
##    Local-gov                88         86        477         27      503
##    Never-worked              0          0          0          0        1
##    Private                 729       1005       3551        181     7780
##    Self-emp-inc             35         38        273         35      279
##    Self-emp-not-inc         71        108        399         50      866
##    State-gov                41         46        270         89      268
##    Without-pay               1          0          0          0        9
##
##                     Masters  Preschool  Prof-school  Some-college
##    ?                     48          5           18           514
##    Federal-gov           67          0           29           254
##    Local-gov            342          4           29           387
##    Never-worked           0          0            0             2
##    Private              894         41          257          5094
##    Self-emp-inc          79          0           81           226
##    Self-emp-not-inc     124          0          131           486
##    State-gov            169          1           31           325
##    Without-pay            0          0            0             3
```

```
# Convert to percentages
cross_table_pct <- prop.table(cross_table, margin = 2) * 100

# Create the bar plot
barplot(cross_table_pct, beside = TRUE, col = rainbow(nrow(cross_table)),
        main = "Relationship between Workclass and Education",
        xlab = "Workclass", ylab = "Percentage", las = 2)

# Add a legend
legend("topright", legend = rownames(cross_table), fill = rainbow(nrow(cross_table)), cex = 0.8)
```




```
# 2(a)
setwd("C:/Users/SAM/OneDrive/Documents/data science/as 1")

population_even <- read.csv("population_even.csv")
population_odd <- read.csv("population_odd.csv")

population_both <- left_join(population_even, population_odd, by = "NAME")

head(population_both)
```

```
##   STATE.x       NAME POPESTIMATE2010 POPESTIMATE2012 POPESTIMATE2014
## 1       1    Alabama         4785437         4815588         4841799
## 2       2     Alaska          713910          730443          736283
## 3       4    Arizona         6407172         6554978         6730413
## 4       5   Arkansas         2921964         2952164         2967392
## 5       6 California        37319502        37948800        38596972
## 6       8   Colorado         5047349         5192647         5350101
##   POPESTIMATE2016 POPESTIMATE2018 STATE.y POPESTIMATE2011 POPESTIMATE2013
## 1         4863525         4887681       1         4799069         4830081
## 2          741456          735139       2          722128          737068
## 3         6941072         7158024       4              NA         6632764
## 4         2989918         3009733       5         2940667         2959400
## 5        39167117        39461588       6        37638369        38260787
## 6         5539215         5691287       8         5121108         5269035
##   POPESTIMATE2015 POPESTIMATE2017 POPESTIMATE2019
## 1         4852347         4874486         4903185
## 2          737498          739700          731545
## 3         6829676         7044008         7278717
## 4         2978048         3001345         3017804
## 5        38918045        39358497        39512223
## 6         5450623         5611885         5758736
```

```
# 2(b)

#(a)
duplicate_col <- duplicated(names(population_both))
if (any(duplicate_col)) {
 population_both <- population_both[, !duplicate_col, drop =
FALSE]
}
head(population_both)
```

```
##   STATE.x       NAME POPESTIMATE2010 POPESTIMATE2012 POPESTIMATE2014
## 1       1    Alabama         4785437         4815588         4841799
## 2       2     Alaska          713910          730443          736283
## 3       4    Arizona         6407172         6554978         6730413
## 4       5   Arkansas         2921964         2952164         2967392
## 5       6 California        37319502        37948800        38596972
## 6       8   Colorado         5047349         5192647         5350101
##   POPESTIMATE2016 POPESTIMATE2018 STATE.y POPESTIMATE2011 POPESTIMATE2013
## 1         4863525         4887681       1         4799069         4830081
## 2          741456          735139       2          722128          737068
## 3         6941072         7158024       4              NA         6632764
## 4         2989918         3009733       5         2940667         2959400
## 5        39167117        39461588       6        37638369        38260787
## 6         5539215         5691287       8         5121108         5269035
##   POPESTIMATE2015 POPESTIMATE2017 POPESTIMATE2019
## 1         4852347         4874486         4903185
## 2          737498          739700          731545
## 3         6829676         7044008         7278717
## 4         2978048         3001345         3017804
## 5        38918045        39358497        39512223
## 6         5450623         5611885         5758736
```

```
#(b)
colnames(population_both) <- gsub("POPESTIMATE", "",
colnames(population_both))
head(population_both)
```

```
##   STATE.x       NAME     2010     2012     2014     2016     2018 STATE.y
## 1       1    Alabama  4785437  4815588  4841799  4863525  4887681       1
## 2       2     Alaska   713910   730443   736283   741456   735139       2
## 3       4    Arizona  6407172  6554978  6730413  6941072  7158024       4
## 4       5   Arkansas  2921964  2952164  2967392  2989918  3009733       5
## 5       6 California 37319502 37948800 38596972 39167117 39461588       6
## 6       8   Colorado  5047349  5192647  5350101  5539215  5691287       8
##       2011     2013     2015     2017     2019
## 1  4799069  4830081  4852347  4874486  4903185
## 2   722128   737068   737498   739700   731545
## 3       NA  6632764  6829676  7044008  7278717
## 4  2940667  2959400  2978048  3001345  3017804
## 5 37638369 38260787 38918045 39358497 39512223
## 6  5121108  5269035  5450623  5611885  5758736
```

```
#(c)
year_columns <- grep("^\\d{4}$", colnames(population_both))
sorted_years <- sort(colnames(population_both)[year_columns])
population_both <- population_both[, c("STATE.x", "NAME",
sorted_years)]
head(population_both)
```

```
##   STATE.x       NAME     2010     2011     2012     2013     2014     2015
## 1       1    Alabama  4785437  4799069  4815588  4830081  4841799  4852347
## 2       2     Alaska   713910   722128   730443   737068   736283   737498
## 3       4    Arizona  6407172       NA  6554978  6632764  6730413  6829676
## 4       5   Arkansas  2921964  2940667  2952164  2959400  2967392  2978048
## 5       6 California 37319502 37638369 37948800 38260787 38596972 38918045
## 6       8   Colorado  5047349  5121108  5192647  5269035  5350101  5450623
##       2016     2017     2018     2019
## 1  4863525  4874486  4887681  4903185
## 2   741456   739700   735139   731545
## 3  6941072  7044008  7158024  7278717
## 4  2989918  3001345  3009733  3017804
## 5 39167117 39358497 39461588 39512223
## 6  5539215  5611885  5691287  5758736
```

```
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 4.3.3
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# Replace missing values with the average of surrounding years
population_filled <- population_both %>%
  mutate(across(`2010`:`2019`, ~ na.fill(.x, "extend")))

# Show the head of the filled data
head(population_filled)
```

```
##   STATE.x       NAME     2010     2011     2012     2013     2014     2015
## 1       1    Alabama  4785437  4799069  4815588  4830081  4841799  4852347
## 2       2     Alaska   713910   722128   730443   737068   736283   737498
## 3       4    Arizona  6407172  1831398  6554978  6632764  6730413  6829676
## 4       5   Arkansas  2921964  2940667  2952164  2959400  2967392  2978048
## 5       6 California 37319502 37638369 37948800 38260787 38596972 38918045
## 6       8   Colorado  5047349  5121108  5192647  5269035  5350101  5450623
##       2016     2017     2018     2019
## 1  4863525  4874486  4887681  4903185
## 2   741456   739700   735139   731545
## 3  6941072  7044008  7158024  7278717
## 4  2989918  3001345  3009733  3017804
## 5 39167117 39358497 39461588 39512223
## 6  5539215  5611885  5691287  5758736
```

```
# 2(d)
# (a)

# Get the maximum population for a single year for each state
max_population_per_state <- population_filled %>%
  rowwise() %>%
  mutate(max_population = max(c_across(`2010`:`2019`), na.rm = TRUE))

max_population_per_state
```

```
## # A tibble: 52 × 13
## # Rowwise:
##    STATE.x NAME   `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
##      <int> <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1       1 Alaba… 4.79e6 4.80e6 4.82e6 4.83e6 4.84e6 4.85e6 4.86e6 4.87e6 4.89e6
##  2       2 Alaska 7.14e5 7.22e5 7.30e5 7.37e5 7.36e5 7.37e5 7.41e5 7.40e5 7.35e5
##  3       4 Arizo… 6.41e6 1.83e6 6.55e6 6.63e6 6.73e6 6.83e6 6.94e6 7.04e6 7.16e6
##  4       5 Arkan… 2.92e6 2.94e6 2.95e6 2.96e6 2.97e6 2.98e6 2.99e6 3.00e6 3.01e6
##  5       6 Calif… 3.73e7 3.76e7 3.79e7 3.83e7 3.86e7 3.89e7 3.92e7 3.94e7 3.95e7
##  6       8 Color… 5.05e6 5.12e6 5.19e6 5.27e6 5.35e6 5.45e6 5.54e6 5.61e6 5.69e6
##  7       9 Conne… 3.58e6 3.59e6 3.59e6 3.59e6 3.59e6 3.59e6 3.58e6 3.57e6 3.57e6
##  8      10 Delaw… 9.00e5 9.07e5 9.15e5 9.24e5 9.32e5 9.41e5 9.49e5 9.57e5 9.65e5
##  9      11 Distr… 6.05e5 6.20e5 6.35e5 6.51e5 6.62e5 6.75e5 6.86e5 6.95e5 7.02e5
## 10      12 Flori… 1.88e7 1.91e7 1.93e7 1.95e7 1.98e7 2.02e7 2.06e7 2.10e7 2.12e7
## # ℹ 42 more rows
## # ℹ 2 more variables: `2019` <dbl>, max_population <dbl>
```

```
# (b)
# Get the total population across all years for each state
total_population_per_state <- population_filled %>%
  rowwise() %>%
  mutate(total_population = sum(c_across(`2010`:`2019`), na.rm = TRUE))

total_population_per_state
```

```
## # A tibble: 52 × 13
## # Rowwise:
##    STATE.x NAME   `2010` `2011` `2012` `2013` `2014` `2015` `2016` `2017` `2018`
##      <int> <chr>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1       1 Alaba… 4.79e6 4.80e6 4.82e6 4.83e6 4.84e6 4.85e6 4.86e6 4.87e6 4.89e6
##  2       2 Alaska 7.14e5 7.22e5 7.30e5 7.37e5 7.36e5 7.37e5 7.41e5 7.40e5 7.35e5
##  3       4 Arizo… 6.41e6 1.83e6 6.55e6 6.63e6 6.73e6 6.83e6 6.94e6 7.04e6 7.16e6
##  4       5 Arkan… 2.92e6 2.94e6 2.95e6 2.96e6 2.97e6 2.98e6 2.99e6 3.00e6 3.01e6
##  5       6 Calif… 3.73e7 3.76e7 3.79e7 3.83e7 3.86e7 3.89e7 3.92e7 3.94e7 3.95e7
##  6       8 Color… 5.05e6 5.12e6 5.19e6 5.27e6 5.35e6 5.45e6 5.54e6 5.61e6 5.69e6
##  7       9 Conne… 3.58e6 3.59e6 3.59e6 3.59e6 3.59e6 3.59e6 3.58e6 3.57e6 3.57e6
##  8      10 Delaw… 9.00e5 9.07e5 9.15e5 9.24e5 9.32e5 9.41e5 9.49e5 9.57e5 9.65e5
##  9      11 Distr… 6.05e5 6.20e5 6.35e5 6.51e5 6.62e5 6.75e5 6.86e5 6.95e5 7.02e5
## 10      12 Flori… 1.88e7 1.91e7 1.93e7 1.95e7 1.98e7 2.02e7 2.06e7 2.10e7 2.12e7
## # ℹ 42 more rows
## # ℹ 2 more variables: `2019` <dbl>, total_population <dbl>
```

```
# 2(e)

# Total US population for a single year
total_us_population_2010 <- sum(population_filled$`2010`, na.rm = TRUE)


total_us_population_2010
```

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

```
# 3

# Choose specific states to visualize

# Load required libraries
library(dplyr)
library(tidyr)
library(ggplot2)
library(scales)
```

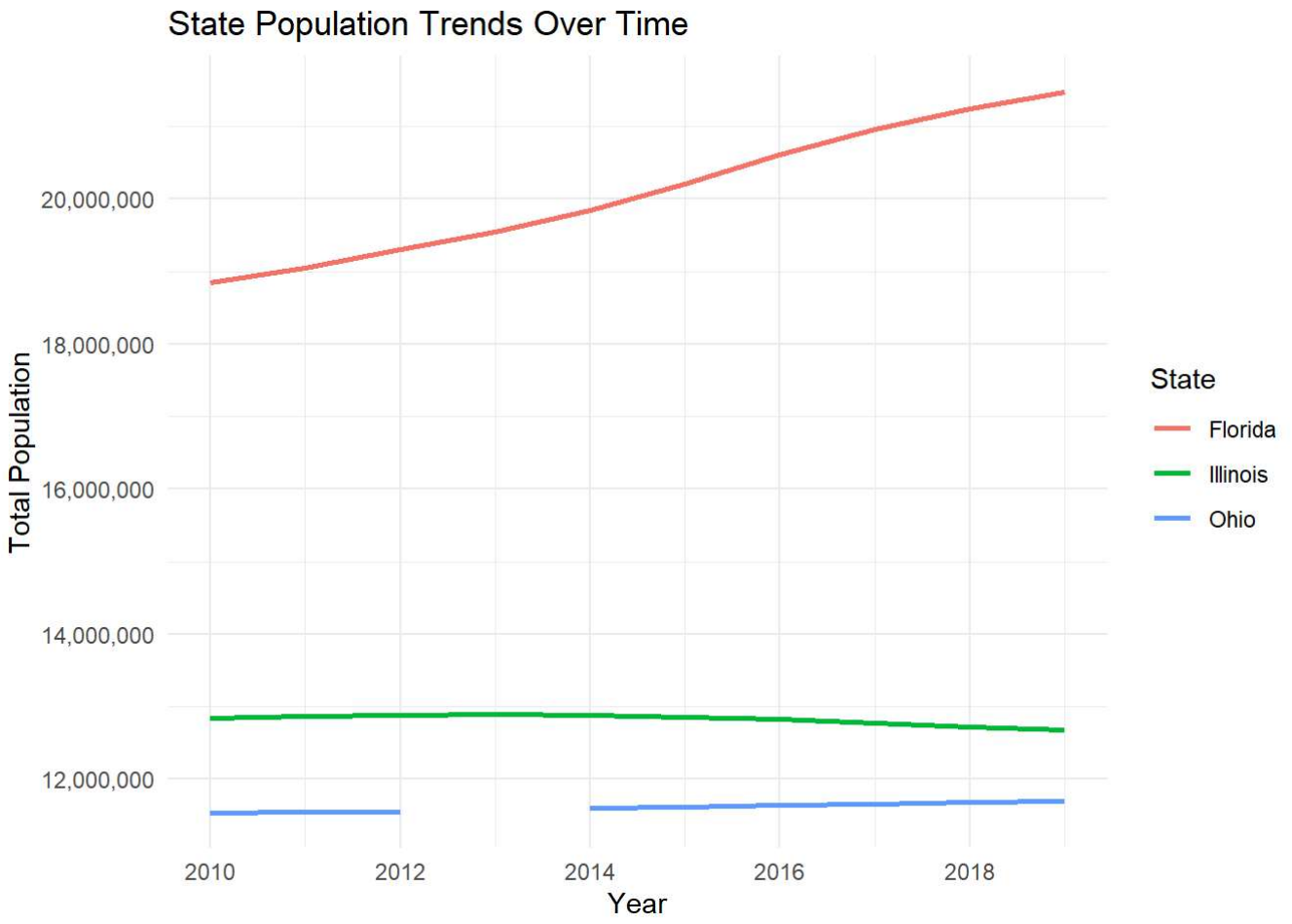
```
## Warning: package 'scales' was built under R version 4.3.3
```

```
##
## Attaching package: 'scales'
```

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

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
# Select specific states
filtered_states <- population_both %>%
  filter(NAME %in% c("Florida", "Illinois", "Ohio"))

# Reshape data to long format
population_tidy <- filtered_states %>%
  pivot_longer(cols = starts_with("20"), names_to = "Year", values_to = "Population")

# Convert year column to numeric format
population_tidy$Year <- as.integer(sub("X", "", population_tidy$Year))

# Create the plot
population_trend <- ggplot(population_tidy, aes(x = Year, y = Population, color = NAME)) +
  geom_line(linewidth = 1) +  # Updated from 'size' to 'linewidth'
  scale_x_continuous(breaks = seq(2010, 2019, by = 2)) +
  scale_y_continuous(labels = comma_format()) +
  labs(
    title = "State Population Trends Over Time",
    x = "Year",
    y = "Total Population",
    color = "State"
  ) +
  theme_minimal()

# Display the plot
print(population_trend)
```


State Population Trends Over Time
Total Population
20,000,000
18,000,000
16,000,000
14,000,000
12,000,000
2010
2012
2014
2016
2018
Year
State
Florida
Illinois
Ohio

#4

#a. Describe two ways in which data can be dirty, and for each one, provide a potential solutio n.
#Ans.
 # Missing data:

#Problem:Missing values in certain records lead to an incomplete analysis.
#Solution:To handle missing values, use predictive modeling, or impute the missing data with the mean or median. Alternatively, remove rows or columns that contain a large number of missing val ues.

 # Inconsistent data:

#Problem: To address missing values, you can either use predictive modeling or impute the missin g data with the mean or median. Alternatively, you may choose to remove rows or columns with a s ignificant amount of missing data.
#Solution:To ensure consistency, standardize the formats across the dataset (e.g., using a consi stent date format). Alternatively, apply string-matching techniques to correct any mismatched te xt entries.

#b.

 # a) Clustering:
#Clustering is used to categorize clients based on similar buying behaviors. For example, K-Mean s or Hierarchical Clustering can identify five distinct client groups that often purchase simila r products.

 # b) Classification:
#By analyzing patterns in historical data, a classification model, such as Decision Trees or Log istic Regression, can predict the probability of a customer buying milk based on their past purc hasing behavior.
#c) Association rule mining:
#Association rule mining identifies product sets that are frequently purchased together, reveali ng relationships such as 'customers who buy bread often buy butter.' This can be achieved using algorithms like FP-Growth or Apriori.

#c.

#a)Organizing a company’s customer base by educational attainment is not a data mining task: I t’s simply categorizing or grouping based on a single feature, without uncovering any hidden pat terns or insights.
#b)Calculating a company’s total sales is not a data mining task: This is a basic aggregation pr ocess that computes a known value, rather than revealing any underlying patterns.
#c)Identifying students in a student database is not a data mining task: Sorting data is a basic organizational activity that does not aim to uncover any significant patterns or insights.
#d) Predicting the outcome of rolling a fair pair of dice is not a data mining task: This involv es a random event, which is not related to discovering patterns or knowledge from data.
#e) Predicting a company's future stock price based on historical data is a data mining task: Th

```
is involves analyzing past data to uncover patterns and make future predictions, often using tec
hniques such as regression or time series analysis.
```

```
```