The data was seperated by tabs and it has 12 columns and 32561 rows. I have replaced all the missing values with NA

```
unclean_data <- read.csv( "data.csv",  sep = "\t", strip.white = TRUE, na.strings =
c("NA", "?"));
#type of data
str(unclean_data)
```

```
## 'data.frame':    32561 obs. of  12 variables:
##  $ X              : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ age            : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass      : Factor w/ 9 levels "Federal-gov",..: 8 7 4 4 4 4 4 7 4 4 ...
##  $ education      : Factor w/ 16 levels "10th","11th",..: 10 10 12 2 10 13 7 12 13
10 ...
##  $ occupation     : Factor w/ 14 levels "Adm-clerical",..: 1 4 6 6 10 4 8 4 10 4
 ...
##  $ 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 ...
##  $ native.country : Factor w/ 43 levels "Cambodia","Canada",..: 41 40 41 41 5 41 2
3 41 41 41 ...
##  $ salaries       : num  43136 46209 28937 33658 34372 ...
##  $ jobsatisfaction: Factor w/ 17 levels "0","1","10","11",..: 1 14 13 13 5 11 14 1
2 5 12 ...
##  $ male           : int  1 1 1 1 NA NA NA 1 NA 1 ...
##  $ female         : int  NA NA NA NA 1 1 1 NA 1 NA ...
```

I have removed Column X because it does not make sense since the rows are always numbered automatically

```
unclean_data[,c("X")] <- NULL
```

1. Analysing column Age

There are 97 people without age value.

```
sum(is.na(unclean_data$age))
```

```
## [1] 97
```

The minimum and maximum age in the data is -57 years and 320 years respectively, I have subsituted these values with missing value because its unrealistic to have negative vale as age neither is it realistic to be over 320 years. i have assumed one can not be over 100 years and below 1 year

```
max(unclean_data$age, na.rm = TRUE)
```

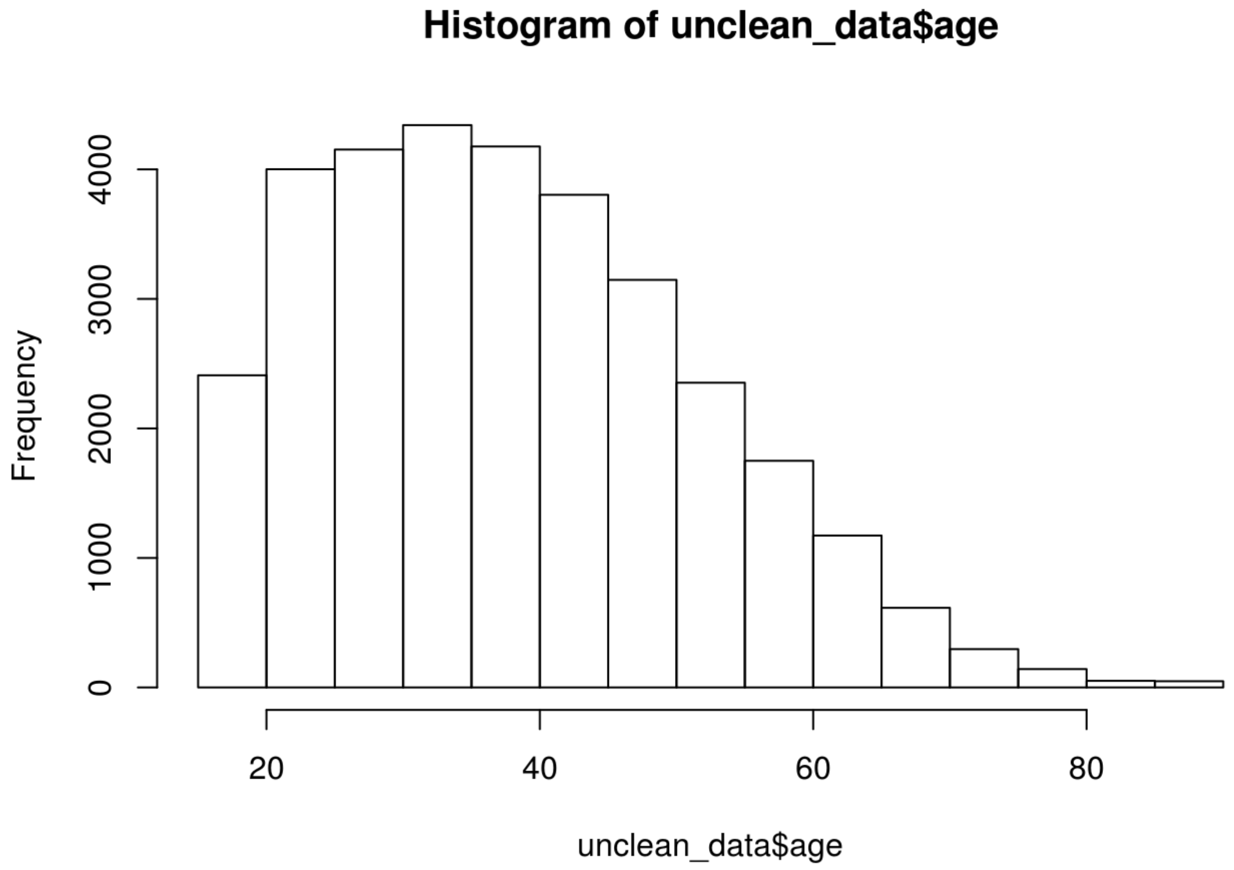
```
## [1] 320
```

```
min(unclean_data$age, na.rm = TRUE)
```

```
## [1] -57
```

```
unclean_data$age[unclean_data$age < 1] <- NA
unclean_data$age[unclean_data$age  >100] <- NA
```

```
hist(unclean_data$age)
```



Age is a ratio

2. Analysing Workclass Column

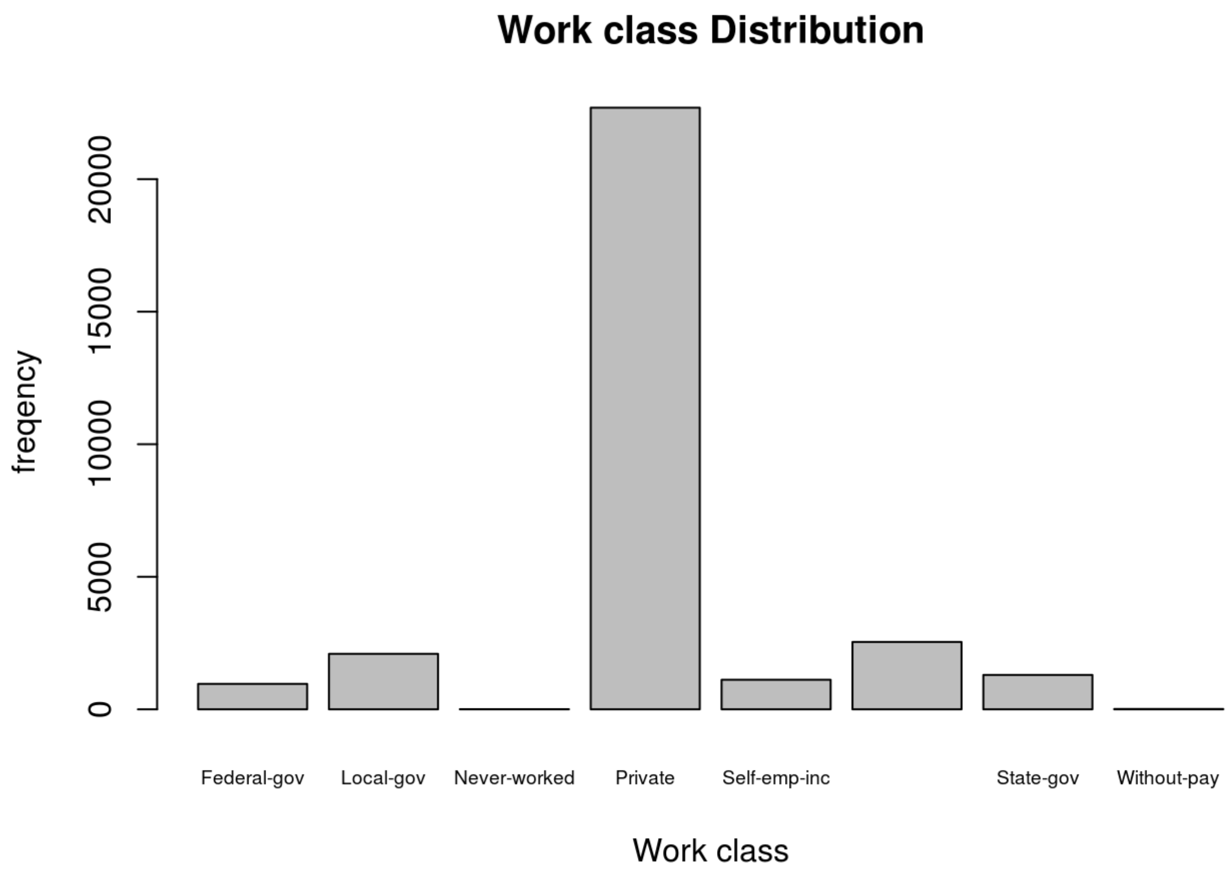
There are 1836 people who are not working.

```
sum(is.na(unclean_data$workclass))
```

```
## [1] 1836
```

There were 10 people who had **privat** as there working class, I decided to add them to **Private** work class. I assumed it was typo since privat was not making sense.

```
library(plyr)
 revalue(unclean_data$workclass, c("privat" = "Private")) -> unclean_data$workclass

counts <- table(unclean_data$workclass)
barplot(counts,  main = "Work class Distribution", xlab = "Work class", ylab = "freqe
ncy", cex.names = 0.6 )
```



Work class is of nominal data type.

3. Analysing Column Education

Everyone atlest went to school. ie there is no missing value

```
library(plyr)
sum(is.na(unclean_data$education))
```

```
## [1] 0
```

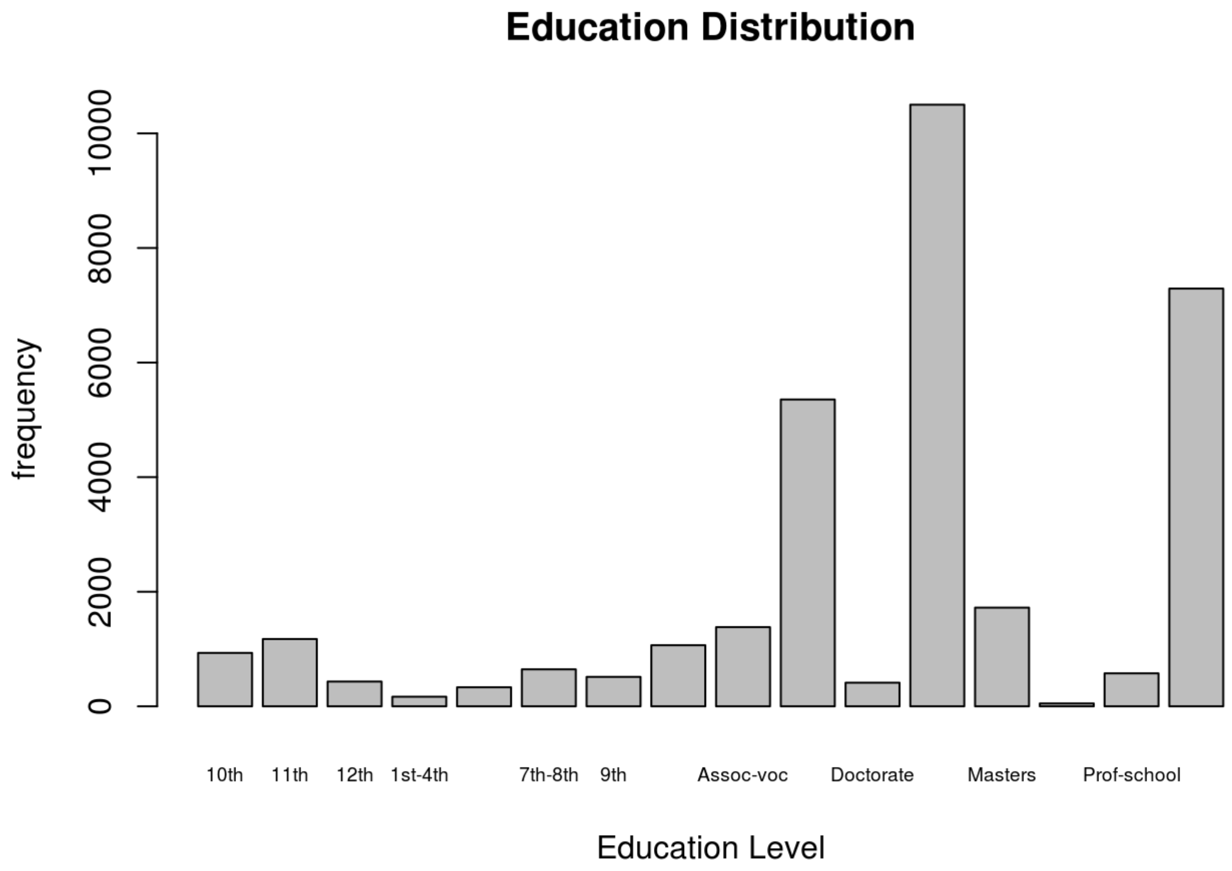
```
counts <-table( unclean_data$education)

unclean_data$education[unclean_data$education =="12th"] <- "1st-12th"
```

```
## Warning in `[<-.factor`(`*tmp*`, unclean_data$education == "12th", value =
## structure(c(10L, : invalid factor level, NA generated
```

```
levels(unclean_data$education)
```

```
##  [1] "10th"         "11th"         "12th"         "1st-4th"
##  [5] "5th-6th"      "7th-8th"      "9th"          "Assoc-acdm"
##  [9] "Assoc-voc"    "Bachelors"    "Doctorate"    "HS-grad"
## [13] "Masters"      "Preschool"    "Prof-school"  "Some-college"
```

```
barplot(counts, main = "Education Distribution", xlab = "Education Level", ylab = "fr
equency", cex.names = 0.6)
```



Education is of ordinal type

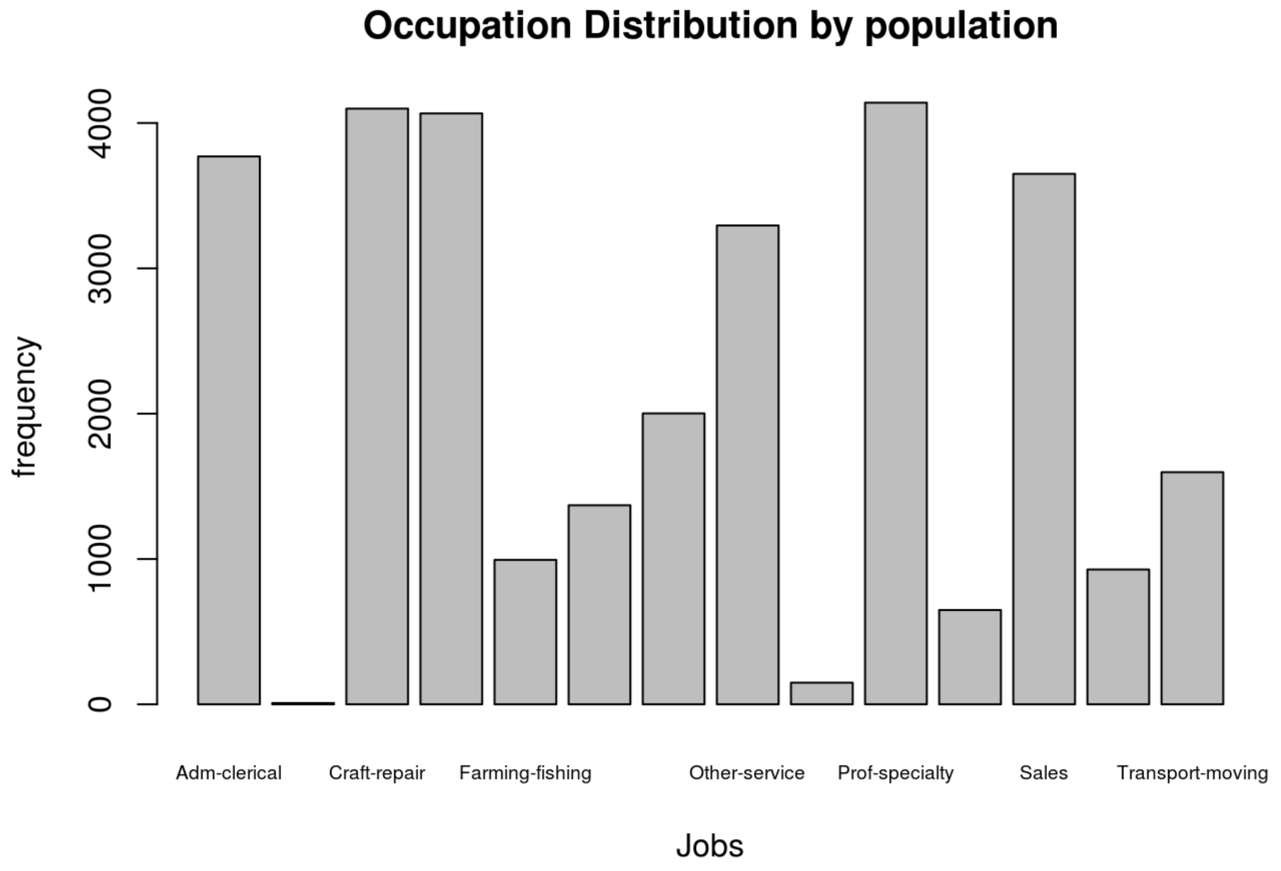
Analysing Attribute occupation

There are 1843 people who do not have occupation

```
sum(is.na(unclean_data$occupation))
```

```
## [1] 1843
```

```
counts <- table(unclean_data$occupation)

barplot(counts,  main = "Occupation Distribution by population", xlab = "Jobs", cex.n
ames = 0.6 ,ylab = "frequency")
```



occupation is of nominal data type.

5. Analysing capital gain

There are no missing value for capital gain. The maximum value is 99999 and the minimum value is 0

```
sum(is.na(unclean_data$capital.gain))
```

```
## [1] 0
```

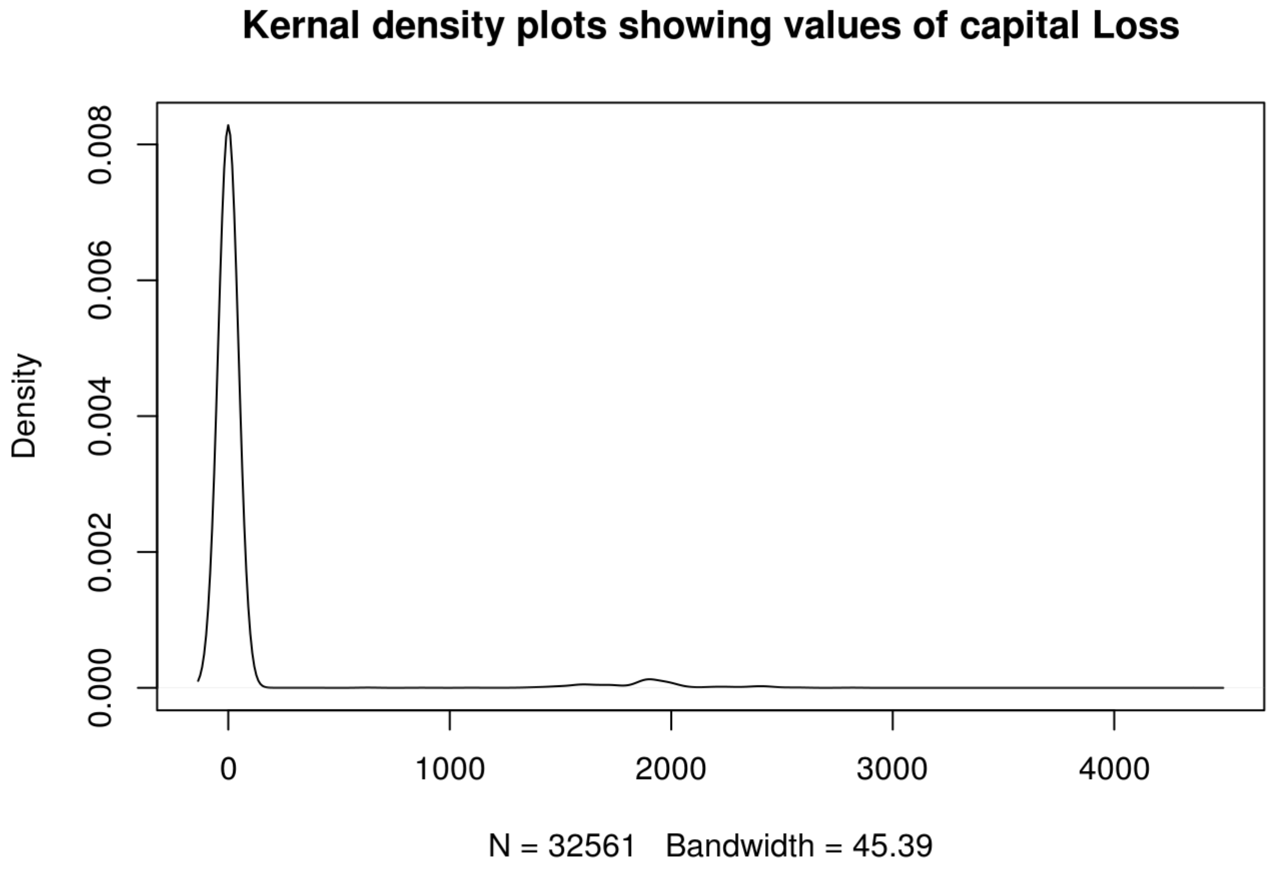
```
max(unclean_data$capital.gain)
```

```
## [1] 99999
```

```
min(unclean_data$capital.gain, na.rm = TRUE)
```

```
## [1] 0
```

```
plot(density(unclean_data$capital.loss), main = "Kernal density plots showing values
 of capital Loss")
```



Capital gain attribute is of ratio data type.

6. Analysing capital loss

There is no missing value for capital loss and the maximum value is 4356 and the minimum value is 0

```
library(plyr, warn.conflicts = FALSE)
sum(is.na(unclean_data$capital.loss))
```

```
## [1] 0
```

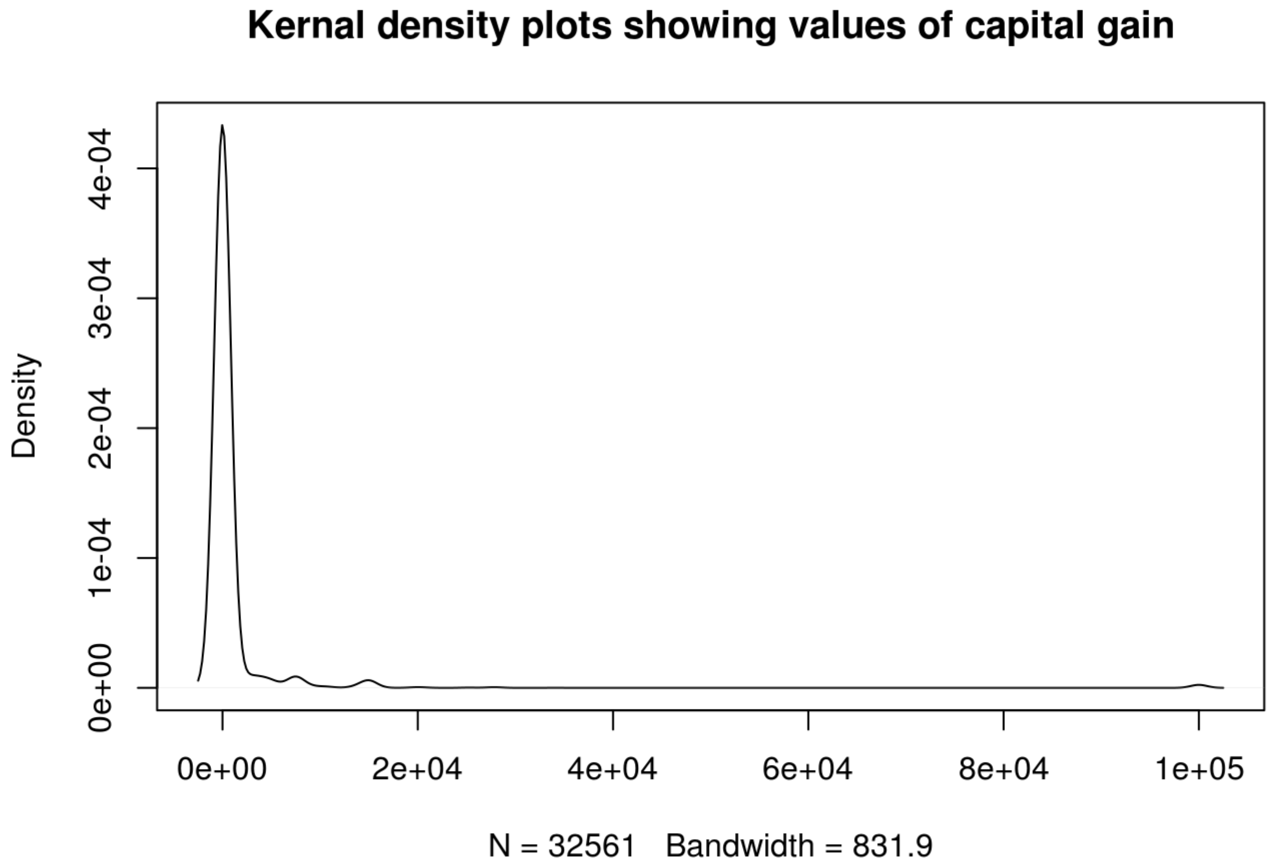
```
max(unclean_data$capital.loss)
```

```
## [1] 4356
```

```
min(unclean_data$capital.loss, na.rm = TRUE)
```

```
## [1] 0
```

```
plot(density(unclean_data$capital.gain), main = "Kernal density plots showing values
 of capital gain")
```



Capital loss attribute is of ratio data type.

7. Analysing column Native country

There are 583 people who are misiing native country

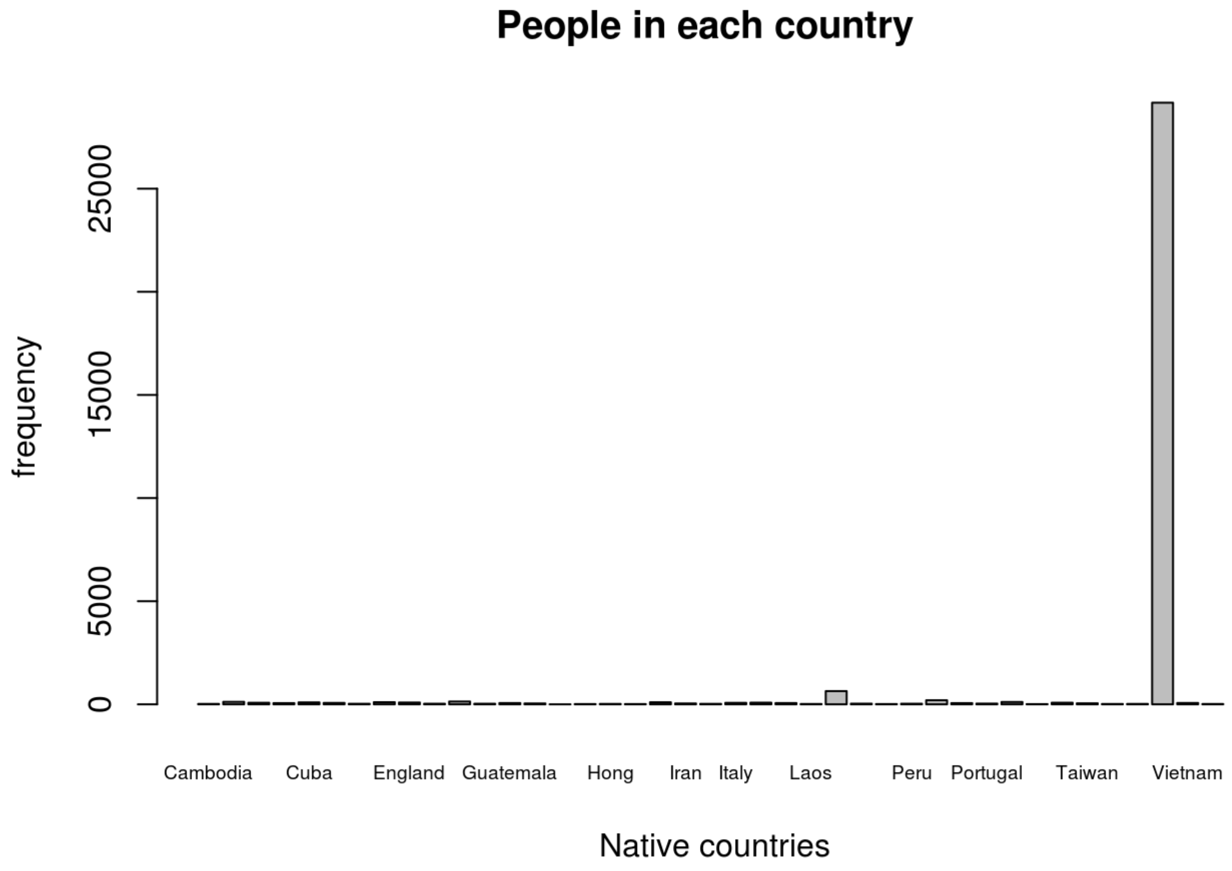
```
sum(is.na(unclean_data$native.country))
```

```
## [1] 583
```

There were 3 countries that shared almost the name ie one was called UnitedStates, Unitedstates and United-States. I replaced UnitedStates and Unitedstates with United-States because there are no countries with such names and I assumed it was a typo that was made.

```
library(plyr)
 revalue( unclean_data$native.country, c("UnitedStates" = "United-States", "Unitedsta
tes" = "United-States" )) ->  unclean_data$native.country
```

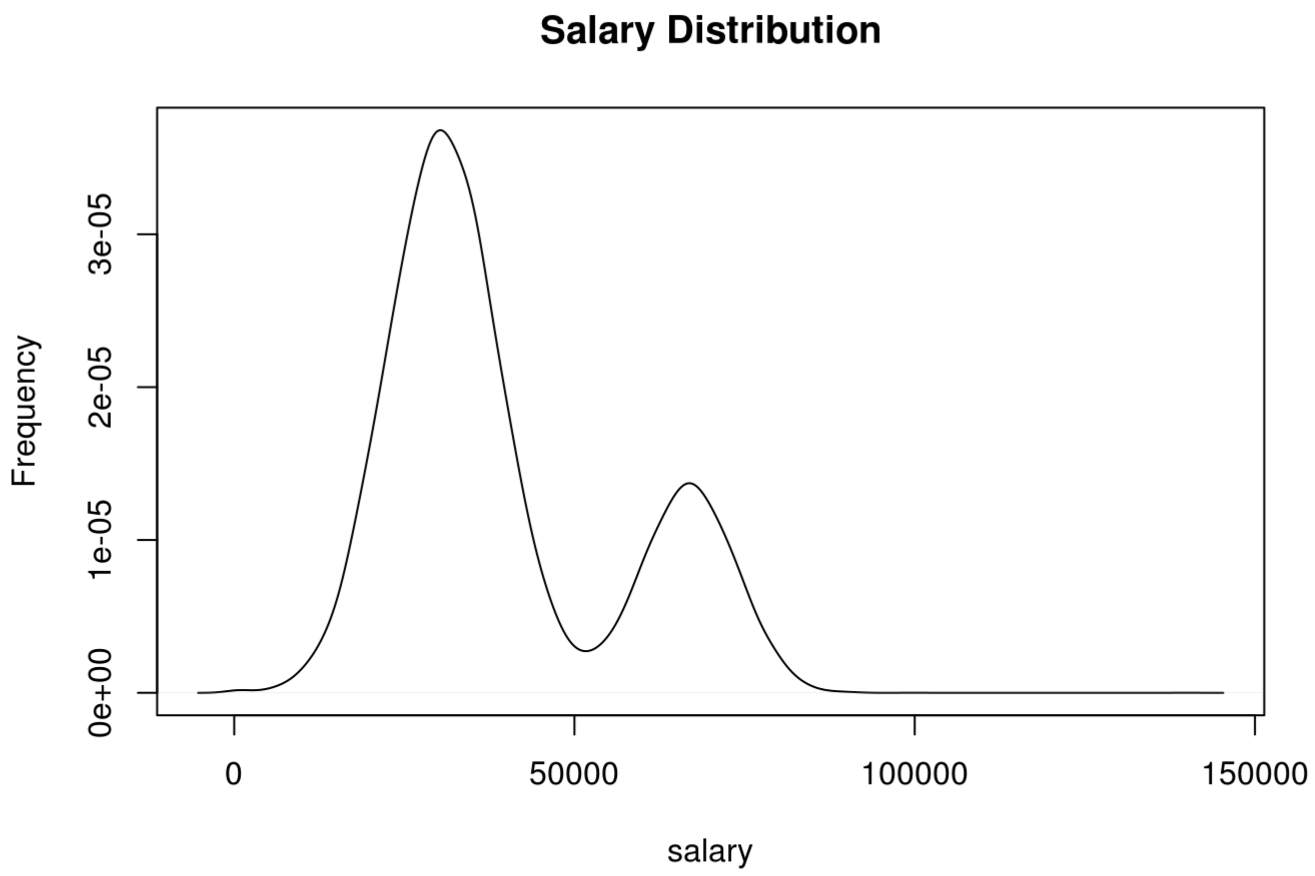
```
counts <- table(unclean_data$native.country)
barplot(counts, main = "People in each country ", xlab = " Native countries", cex.nam
es = 0.6, ylab = "frequency", )
```



8. Analysing column salaries

I rounded off the salary values to 2 decimal places

```
unclean_data$salaries <- format(round(unclean_data$salaries, 2), nsmall = 2)
unclean_data$salaries <- as.numeric(unclean_data$salaries)

plot(density(unclean_data$salaries), main = "Salary Distribution",  xlab = "salary",
ylab = "Frequency")
```



salary is a ratio.

9. Analysing column Jobsatisafction

I changed the value Very good to NA because I think the job satisfaction scale was numeric

```
#levels(as.factor(unclean_data$jobsatisfaction))
#sum(is.na(unclean_data$jobsatisfaction))
#class(unclean_data$jobsatisfaction)

#plot(density(temp.data$jobsatisfaction, na.rm = TRUE), main = "Job satisfaction Dist
ribution")
```

10. Analysing column Male.

1 is used to denote males and there 21790 males

```
data.frame(table(unclean_data$male))
```

```
##   Var1  Freq
## 1    1 21790
```

11. Analysing Column Female

1 is used to analyse females and there 10771 females

```
library(plyr, warn.conflicts = FALSE)

table(unclean_data$female)
```

```
## 
##     1 
## 10771
```

Exercise 3

a) Create a table where each row stands for an occupation, each column stands for a level of education, and the cells in the table contain the average salary of people with the corresponding occupation and education level.

```
library(dplyr, warn.conflicts = FALSE)
library(ggplot2, warn.conflicts = FALSE)
library(tidyr, warn.conflicts = FALSE)
data.occupation.education  = group_by(unclean_data, occupation, education)
data.avg.sal               = summarise(data.occupation.education,
average_salary=mean(salaries))
head(data.avg.sal)
```

```
## # A tibble: 6 x 3
## # Groups:   occupation [1]
##     occupation  education average_salary
##         <fctr>     <fctr>          <dbl>
## 1 Adm-clerical       10th       29957.97
## 2 Adm-clerical       11th       29976.85
## 3 Adm-clerical    5th-6th       26075.00
## 4 Adm-clerical    7th-8th       35226.91
## 5 Adm-clerical        9th       32473.00
## 6 Adm-clerical Assoc-acdm       34836.39
```

```
data.table <- spread(data.avg.sal, key=education, value=average_salary)
data.table
```

```
## # A tibble: 15 x 17
## # Groups:   occupation [15]
##           occupation   `10th`   `11th` `1st-4th` `5th-6th` `7th-8th`
##  *            <fctr>    <dbl>    <dbl>     <dbl>     <dbl>     <dbl>
##  1      Adm-clerical 29957.97 29976.85        NA  26075.00  35226.91
##  2      Armed-Forces       NA       NA        NA        NA        NA
##  3      Craft-repair 35271.21 36050.30  32884.09  33559.63  33048.06
##  4   Exec-managerial 38655.33 36805.44  50014.50  71742.00  42949.74
##  5   Farming-fishing 32535.09 34774.05  31450.00  30814.36  33664.48
##  6 Handlers-cleaners 32047.00 30906.23  31633.06  32748.47  30780.35
##  7 Machine-op-inspct 32183.33 31165.66  32227.96  33421.93  33776.59
##  8     Other-service 30484.64 31202.76  29770.65  31433.08  31001.63
##  9   Priv-house-serv 33172.00 27458.43  25066.36  26133.07  28921.25
## 10    Prof-specialty 45071.67 30745.05  23128.50  39091.00  28300.22
## 11   Protective-serv 26930.17 38363.14  41685.00  19378.00  30010.33
## 12             Sales 33370.27 31411.95  31324.25  39858.67  37247.90
## 13      Tech-support 44464.67 32796.67        NA  39695.00  32947.20
## 14  Transport-moving 38104.95 32254.36  33409.88  31557.00  35341.45
## 15              <NA> 31292.25 29582.05  30193.58  31864.57  32282.27
## # ... with 11 more variables: `9th` <dbl>, `Assoc-acdm` <dbl>,
## #   `Assoc-voc` <dbl>, Bachelors <dbl>, Doctorate <dbl>, `HS-grad` <dbl>,
## #   Masters <dbl>, Preschool <dbl>, `Prof-school` <dbl>,
## #   `Some-college` <dbl>, `<NA>` <dbl>
```

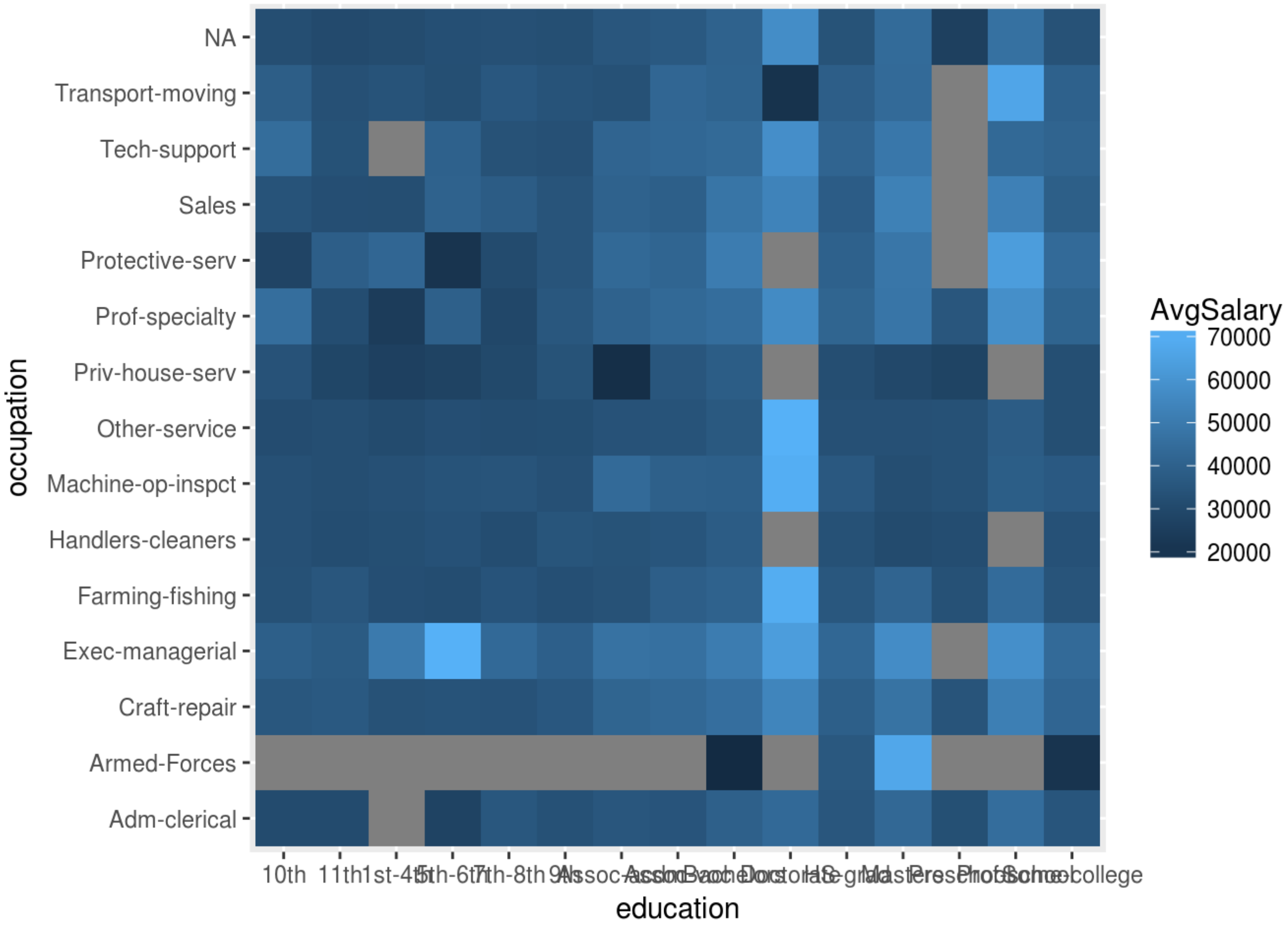
b

```
data.long <- gather(data.table, education, AvgSalary, "10th":"Some-college")
data.long
```

```
## # A tibble: 225 x 4
## # Groups:   occupation [15]
##           occupation   `<NA>` education AvgSalary
##               <fctr>    <dbl>     <chr>     <dbl>
##  1      Adm-clerical 29722.61      10th  29957.97
##  2      Armed-Forces 48635.00      10th        NA
##  3      Craft-repair 35800.19      10th  35271.21
##  4   Exec-managerial 36365.31      10th  38655.33
##  5   Farming-fishing 33999.88      10th  32535.09
##  6 Handlers-cleaners 32224.45      10th  32047.00
##  7 Machine-op-inspct 28159.49      10th  32183.33
##  8     Other-service 31881.76      10th  30484.64
##  9   Priv-house-serv 34069.25      10th  33172.00
## 10    Prof-specialty 37524.00      10th  45071.67
## # ... with 215 more rows
```

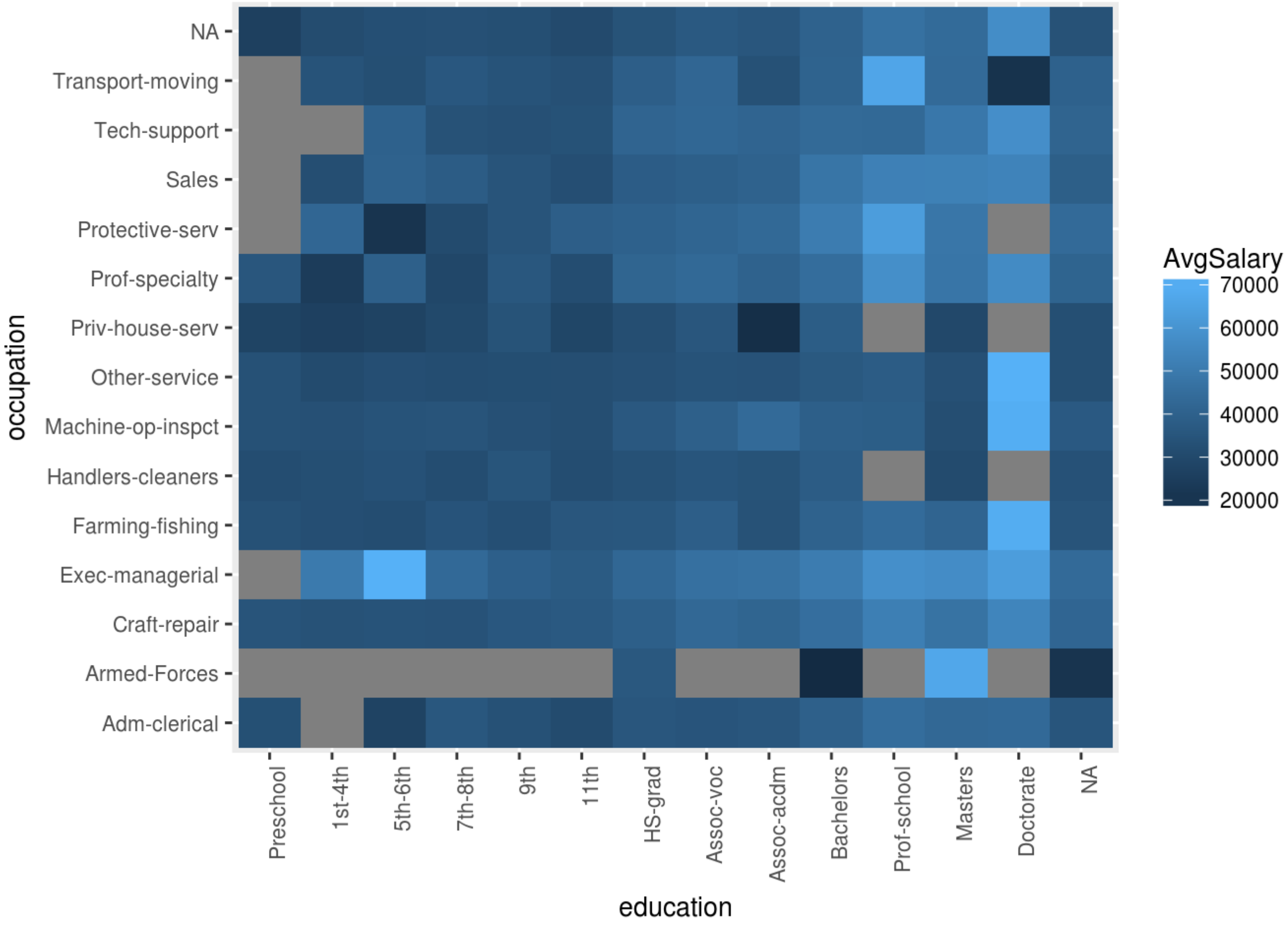
c)

```
library(ggplot2, warn.conflicts = FALSE)

ggplot(data.long,  aes(x=education, y=occupation)) +  geom_tile(aes(x=education, y=oc
cupation, fill=AvgSalary))
```



d)

```
#data <- data.frame(lapply(data.long, factor))


data.long$education <- factor(data.long$education, levels = c("Preschool", "1st-4th",
 "5th-6th", "7th-8th", "9th", "11th", "12th", "HS-grad", "Assoc-voc",  "Assoc-acdm",
"levels",
                                                           "Bachelors", "Prof-school",
 "Masters", "Doctorate" ) , ordered = TRUE)


ggplot(data.long,  aes(x=education, y=occupation)) +  geom_tile(aes(x=education, y=oc
cupation, fill=AvgSalary)) +theme(text = element_text(size=10),
        axis.text.x = element_text(angle=90, hjust=1))
```



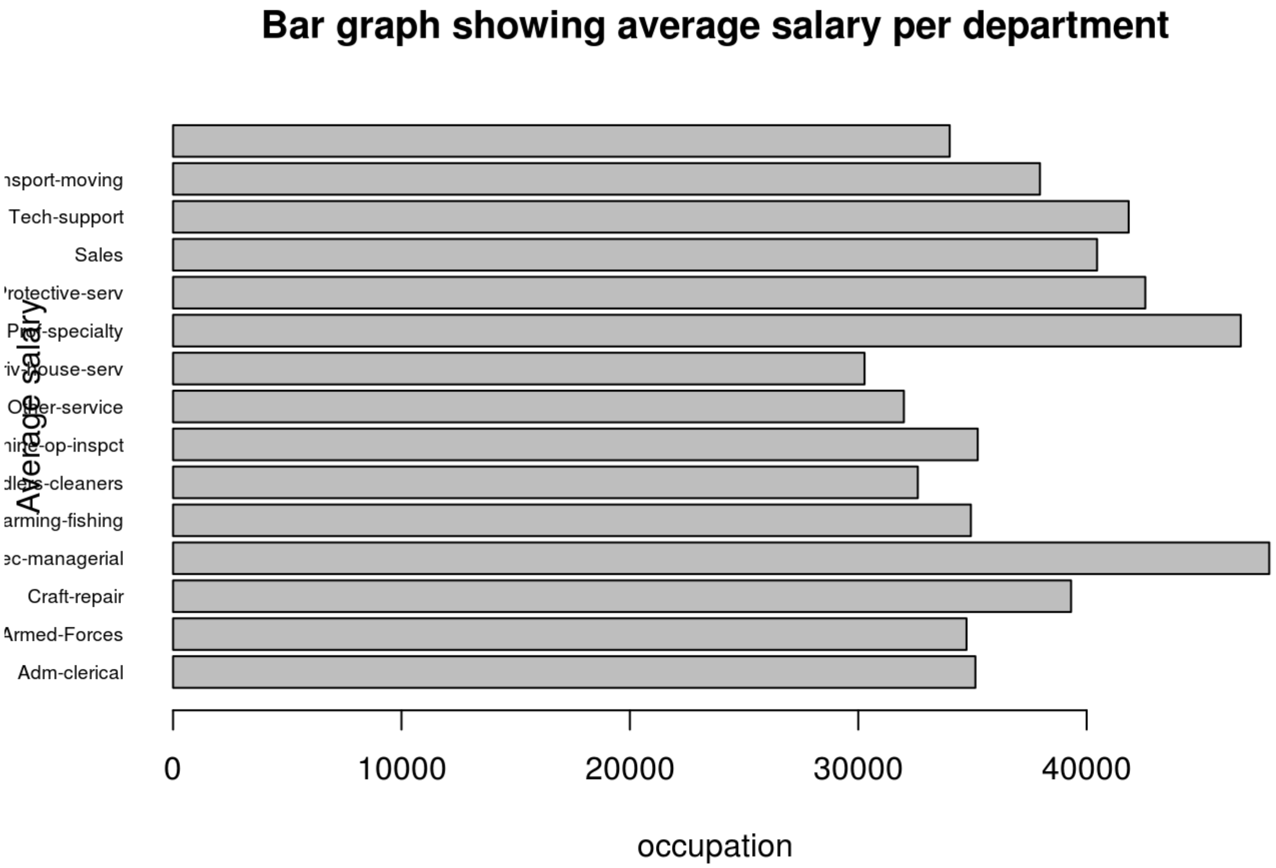
e) List 3 interesting facts that you can read out from this plot.

-Those who ended in preschool can not work in armed forses, protective services, sales, Tech-suppor and transport. -Those who have masters they atlest everyone has a job.

f) Create another plot of this dataset that you think conveys interesting information

I first grouped the data by occupation and the i computed the average salary of each occupation using mean function

```
grouped_data <- group_by(unclean_data, occupation)
summ_data <- summarise(grouped_data, AvgSalary=mean(salaries))
barplot(summ_data$AvgSalary, names.arg=summ_data$occupation, horiz = TRUE, las=1, ce
x.names = 0.6, xlab = "occupation", ylab = "Average salary", main = "Bar graph showin
g average salary per department")
```



- on average those in the exec-managerial and prof-speciality department earn more than any other department. -Those in the riv-house-serv department earn the least followed by those in the Handlers-cleaners and those in other service department. -The income difference between departments is not high.

Exe4

a

The data has 100000 rows and 15 attributes

```
df <- read.csv("instacart.csv")
str(df)
```

```
## 'data.frame':    100000 obs. of  15 variables:
##  $ order_id              : int  2539329 2539329 2539329 2539329 2539329 2398795 23
98795 2398795 2398795 2398795 ...
##  $ user_id               : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ eval_set              : Factor w/ 3 levels "prior","test",..: 1 1 1 1 1 1 1 1 1
1 ...
##  $ order_number          : int  1 1 1 1 1 2 2 2 2 2 ...
##  $ order_dow             : int  2 2 2 2 2 3 3 3 3 3 ...
##  $ order_hour_of_day     : int  8 8 8 8 8 7 7 7 7 7 ...
##  $ days_since_prior_order: int  NA NA NA NA NA 15 15 15 15 15 ...
##  $ product_id            : int  196 14084 12427 26088 26405 196 10258 12427 13176
 26088 ...
##  $ add_to_cart_order     : int  1 2 3 4 5 1 2 3 4 5 ...
##  $ reordered             : int  0 0 0 0 0 1 0 1 0 1 ...
##  $ product_name          : Factor w/ 12571 levels "0 Calorie Fuji Apple Pear Water
Beverage",..: 10555 8310 8492 336 12459 10555 9112 8492 742 336 ...
##  $ aisle_id              : int  77 91 23 23 54 77 117 23 24 23 ...
##  $ department_id         : int  7 16 19 19 17 7 19 19 4 19 ...
##  $ aisle                 : Factor w/ 134 levels "air fresheners candles",..: 118 1
20 104 104 100 118 89 104 51 104 ...
##  $ department            : Factor w/ 21 levels "alcohol","babies",..: 4 8 21 21 12
4 21 21 20 21 ...
```

order_id Is of nominal data type. It describes the order of each client who buys a product.

user_id is a nominal since its used to identify each person.

eval_set

```
levels(df$eval_set)
```

```
## [1] "prior" "test"  "train"
```

```
levels(as.factor(df$order_dow))
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

its of Nominal attribute.

order_number its a ratio

```
levels(as.factor(df$order_number))
```

```
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11" 
##  [12] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22" 
##  [23] "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33" 
##  [34] "34"  "35"  "36"  "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44" 
##  [45] "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55" 
##  [56] "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66" 
##  [67] "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77" 
##  [78] "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88" 
##  [89] "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99" 
## [100] "100"
```

order_dow its a ratio because the values are integer values.

```
levels(as.factor(df$order_dow))
```

```
## [1] "0" "1" "2" "3" "4" "5" "6"
```

order_hour_of_day its a an ordinal value

```
levels(as.factor(df$order_hour_of_day))
```

```
##  [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"
## [15] "14" "15" "16" "17" "18" "19" "20" "21" "22" "23"
```

** days_since_prior_order** its an interval

```
levels(as.factor(df$days_since_prior_order))
```

```
##  [1] "0"  "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10" "11" "12" "13"
## [15] "14" "15" "16" "17" "18" "19" "20" "21" "22" "23" "24" "25" "26" "27"
## [29] "28" "29" "30"
```

product_id its a nominal value because it uniquely identifies each product

add_to_cart_order its a ratio

reordered its an ordinal value.

product_name its a nominal value because it uniquely identifies the product

department_id its

```
levels(as.factor(df$department))
```

```
##  [1] "alcohol"         "babies"          "bakery"
##  [4] "beverages"       "breakfast"       "bulk"
##  [7] "canned goods"    "dairy eggs"      "deli"
## [10] "dry goods pasta" "frozen"          "household"
## [13] "international"   "meat seafood"    "missing"
## [16] "other"           "pantry"          "personal care"
## [19] "pets"            "produce"         "snacks"
```

aisle its a nominal value that uniquely identifies each aisle

```
levels(as.factor(df$aisle_id))
```

```
##   [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11" 
##  [12] "12"  "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22" 
##  [23] "23"  "24"  "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33" 
##  [34] "34"  "35"  "36"  "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44" 
##  [45] "45"  "46"  "47"  "48"  "49"  "50"  "51"  "52"  "53"  "54"  "55" 
##  [56] "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"  "65"  "66" 
##  [67] "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77" 
##  [78] "78"  "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88" 
##  [89] "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99" 
## [100] "100" "101" "102" "103" "104" "105" "106" "107" "108" "109" "110"
## [111] "111" "112" "113" "114" "115" "116" "117" "118" "119" "120" "121"
## [122] "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
## [133] "133" "134"
```

- **aisle** its a nominal value because each aisle has its own unique name

department its a nominal value because each department has its own name

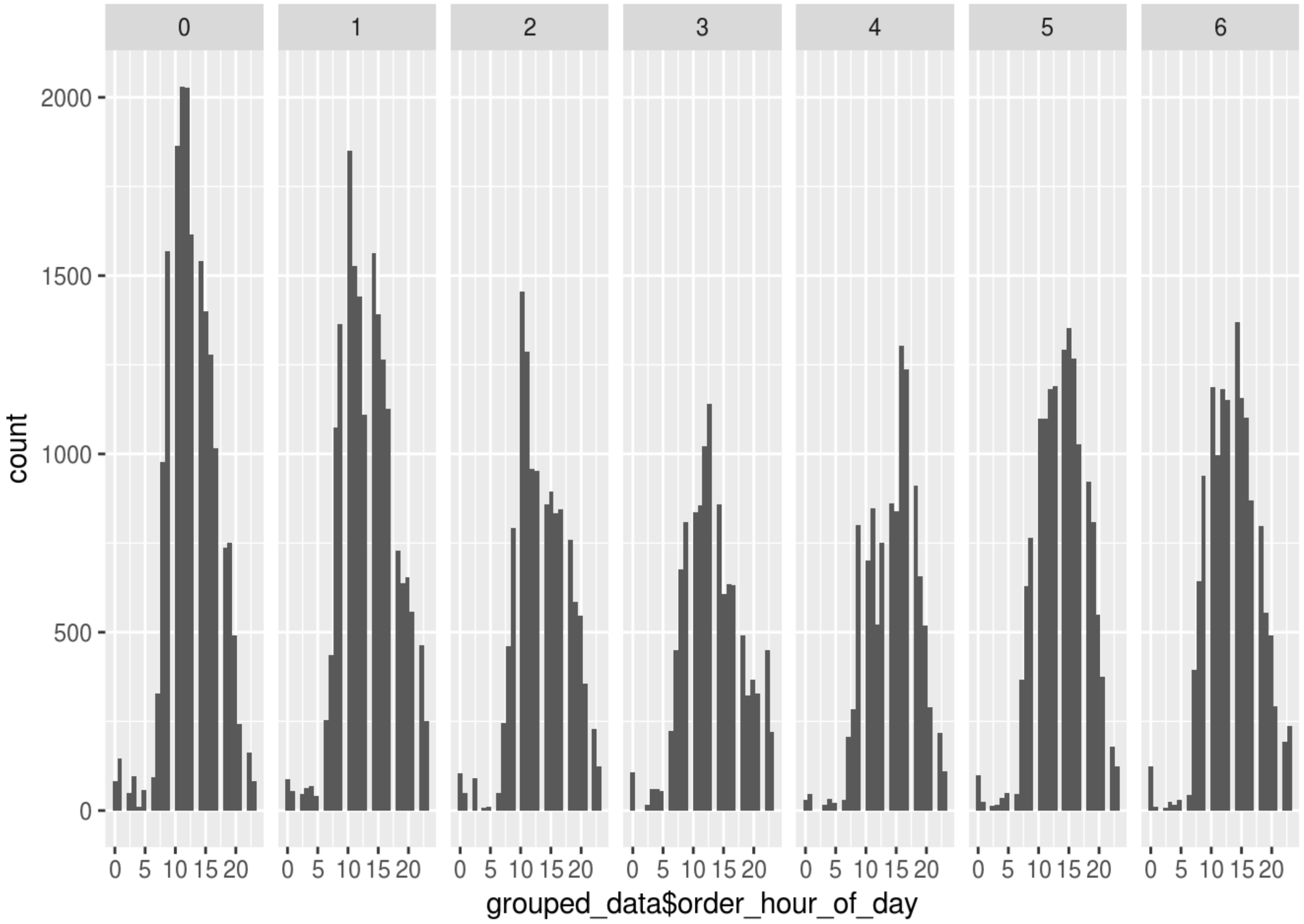
```
levels(df$department)
```

```
##  [1] "alcohol"         "babies"          "bakery"         
##  [4] "beverages"       "breakfast"       "bulk"           
##  [7] "canned goods"    "dairy eggs"      "deli"           
## [10] "dry goods pasta" "frozen"          "household"      
## [13] "international"   "meat seafood"    "missing"        
## [16] "other"           "pantry"          "personal care"  
## [19] "pets"            "produce"         "snacks"
```

b

```
library(ggplot2, warn.conflicts = FALSE)
grouped_data <- group_by(df, df$order_hour_of_day)

ggplot(grouped_data, aes(x=grouped_data$order_hour_of_day))+ geom_histogram() + facet
_grid(. ~grouped_data$order_dow)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



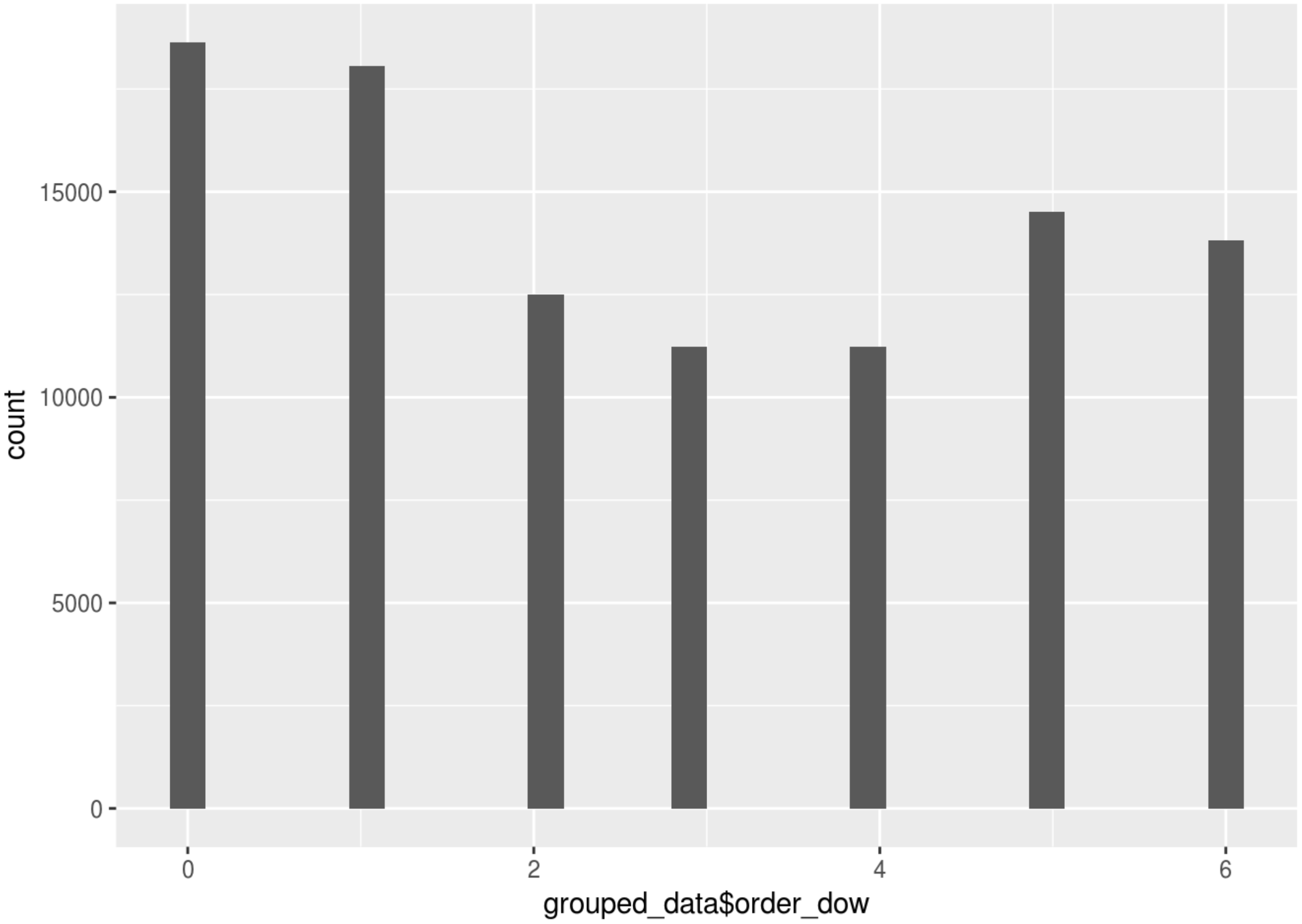
##c) List 3 interesting facts that you can read out from this plot. -its evident from the histogram that between 10 and 15 hours , there is high turn over of sales for the entire week and sunday having the highes turn over with over 2000 clients.

- its also evident from the histogram that the least sales in the week are made between 0 and 5 hours.
- its also evident that all the days of the week the sales tend take a similar patter.

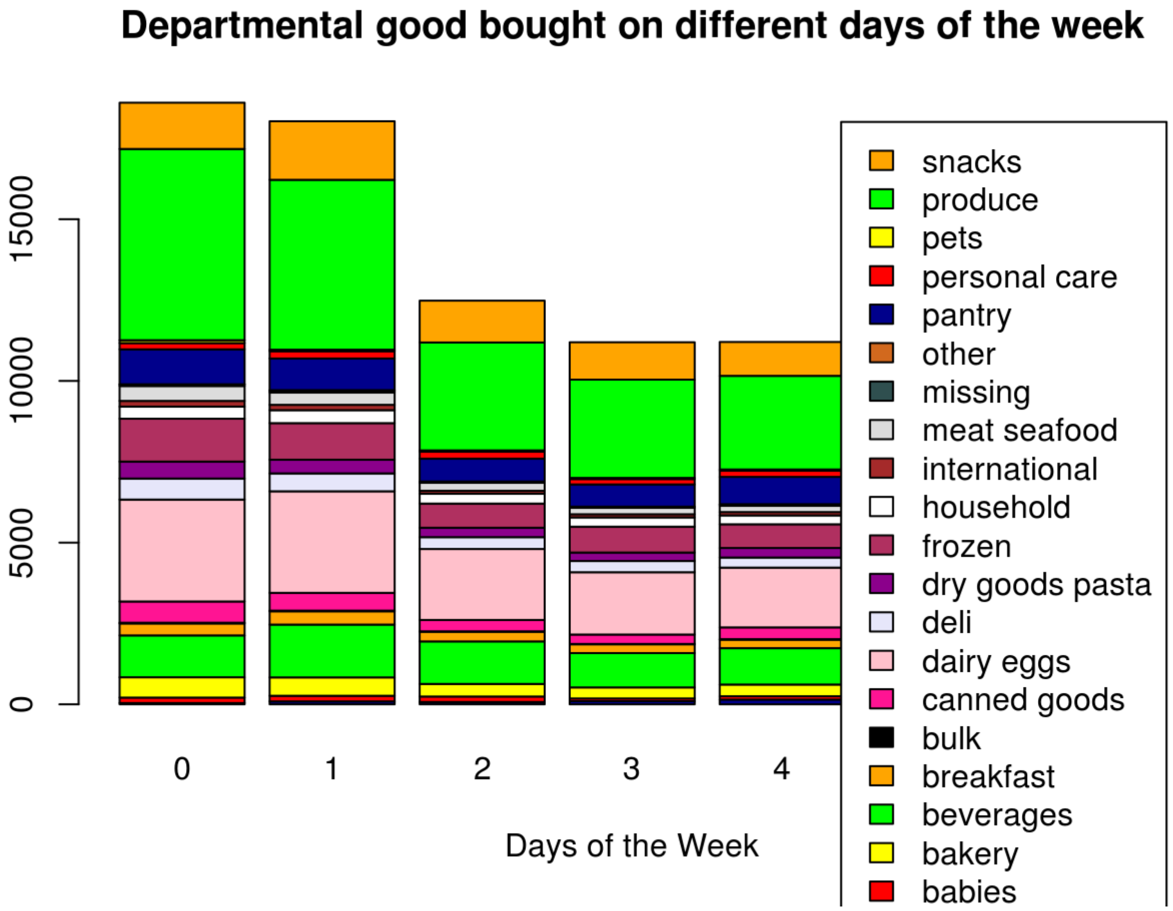
d)

```
grouped_data <- group_by(df, df$order_dow, df$department)

ggplot(grouped_data, aes(x=grouped_data$order_dow))+ geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
counts <- table( df$department, df$order_dow)
barplot(counts, main="Departmental good bought on different days of the week",
  xlab="Days of the Week", col=c("darkblue","red","yellow","green","orange",
"black","DeepPink","Pink", "Lavender","DarkMagenta","Maroon","white","brown","Gainsbo
ro","DarkSlateGray","Chocolate"),
    legend = rownames(counts))
```



-from the stacked bar graph its obvious that the produce department has the highest sells through the week.

-diary eggs department has the second highest sells during the course of the weeek

-bulk department has the least sells through the week.