Census Data Classification & Analysis Using Decision Tree

Abstract: The assigned milestone focuses on examining the given dataset and aims to analyze the dataset to predict whether an individual's income is > 50,000 USD or <= 50,000 USD based on several attributes from the dataset, using decision tree technique.

Introduction: This document details the steps involved in analyzing the census data. The census data includes a total of 32,561 rows.
Steps 1 to 5 explain the data analysis process in detail. First, the features of the dataset are described and the inferences are recorded. Next, the importance and process of cleaning the dataset is described. Third, the processes of obtaining a sample from dataset and splitting into train and test data is explained. In the fourth step, decision tree algorithm with entropy and information gain is explained along with the details of how the best split is chosen by the tree. Finally, the building of the decision tree is detailed.

STEPS INVOLVED IN THE ANALYSIS

Step 1: Exploring the dataset

The given dataset is studied and it is found that it has 32,561 entries/rows and 15 columns. Each attribute/column gives the following information.
age: the age of an individual; type-integer > 0
workclass: the employment status of an individual; type-Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked
fnlwgt: final weight; type-integer > 0
education: the highest level of education pursued by an individual; type- Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool
education-num: the highest level of education in the numerical form; type-integer > 0
marital-status: marital-status of an individual; type- Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse
occupation: the occupation of an individual; type- Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces

relationship: how the individual is related to others; type-Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried
race: the race of an individual; type - White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black
sex: the gender of an individual; type - Male, female
capital-gain: the capital gains of an individual; type- Integer >= 0
capital-loss: the capital losses of an individual; type- Integer >= 0
hours-per-week: the no. of hours an individual works in a week; type- Integer>=0
native-country: the country to which an individual belongs to; type-United States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam,Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands
income: whether an individual earns >50k USD or <=50k USD

We are interested in finding whether an individual earns > 50,000 USD or <= 50,000 USD, considering the above features/attributes.
Here, every feature is analyzed to predict how many individuals earn <= 50,000 USD or > 50,000 USD. The technical and graphical representations of each feature are given below.

Continuous Variables:

There are 5 attributes that are continuous- age, education_num, capital_gain, capital_loss, hours_per_week.

1. Analysis of Age against Income

```
table (train[,c("age", "income")])
```

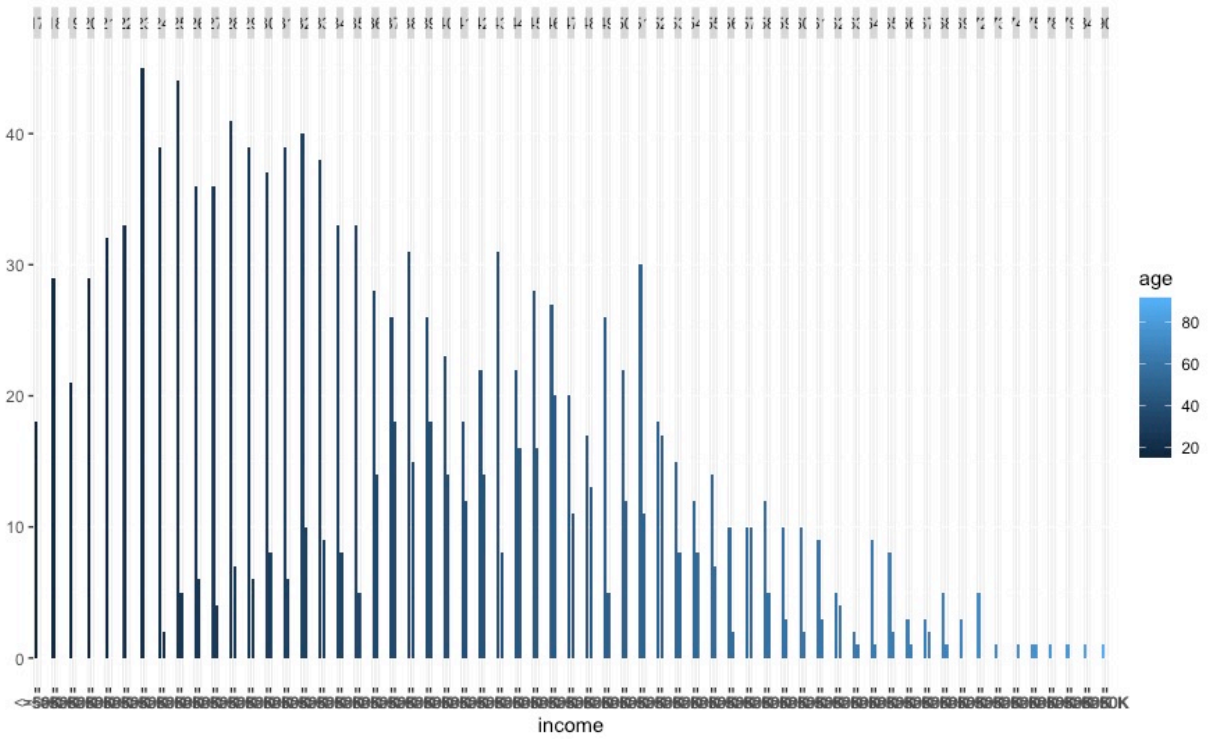
This function returns the distribution of income feature against age.

```
        income
age   <=50K  >50K
  17    18     0
  18    29     0
  19    21     0
  20    29     0
  21    32     0
  22    33     0
```

23	45	0
24	39	2
25	44	5
26	36	6
27	36	4
28	41	7
29	39	6
30	37	8
31	39	6
32	40	10
33	38	9
34	33	8
35	33	5
36	28	14
37	26	18
38	31	15
39	26	18
40	23	14
41	18	12
42	22	14
43	31	8
44	22	16
45	28	16
46	27	20
47	20	11
48	17	13
49	26	5
50	22	12
51	30	11
52	18	17
53	15	8
54	12	8
55	14	7
56	10	2
57	10	10
58	12	5
59	10	3
60	10	2
61	9	3
62	5	4

63	2	1
64	9	1
65	8	2
66	3	1
67	3	2
68	5	1
69	3	0
72	5	0
73	1	0
74	0	1
75	1	1
78	1	0
79	1	0
84	1	0
90	1	0

qplot (income, data = train, fill = age) + facet_grid (. ~ age)



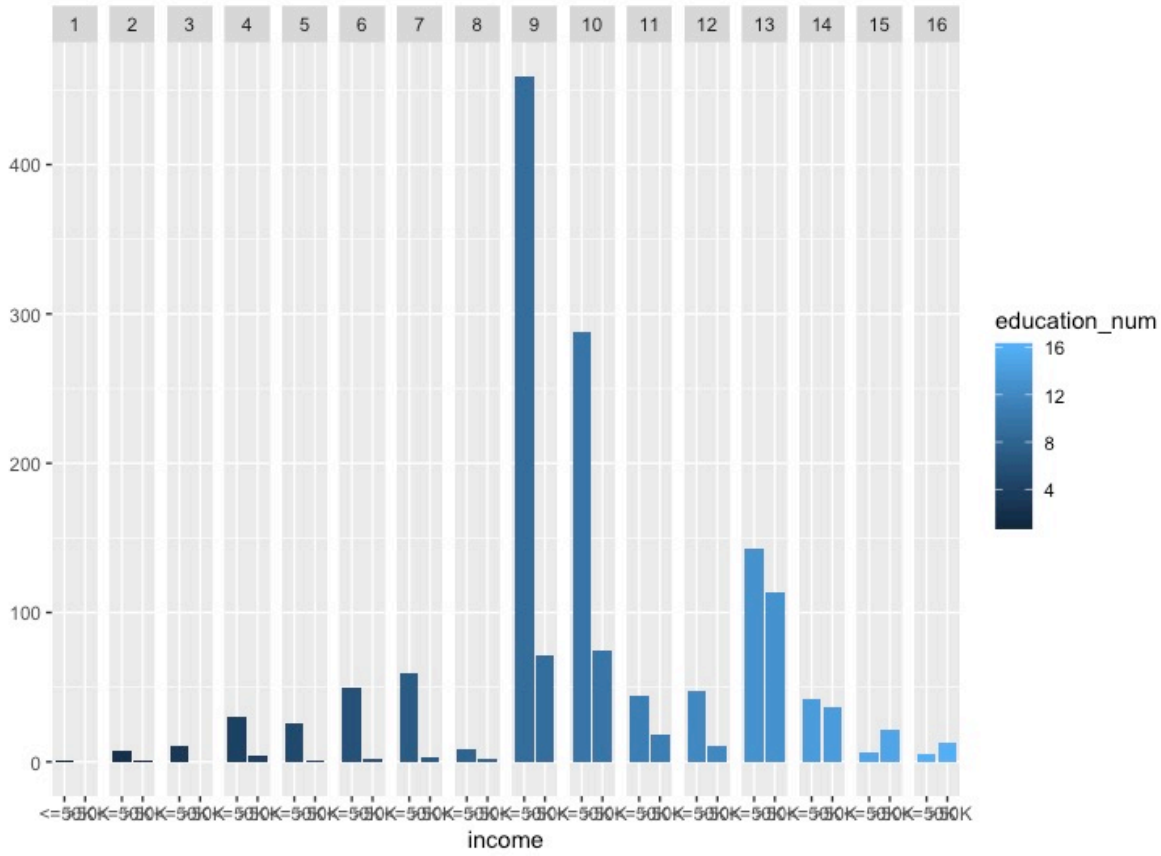
2. Analysis of education_num against Income

table (train[,c(“education_num", "income")])

This function returns the distribution of income feature against education_num.

education_num	income <=50K	>50K
1	1	0
2	7	1
3	11	0
4	30	4
5	26	1
6	50	2
7	59	3
8	9	2
9	459	71
10	288	75
11	44	18
12	48	11
13	143	113
14	42	37
15	6	21
16	5	13

qplot (income, data = train, fill = education_num) + facet_grid (. ~ education_num)



3. Analysis of capital_gain against Income

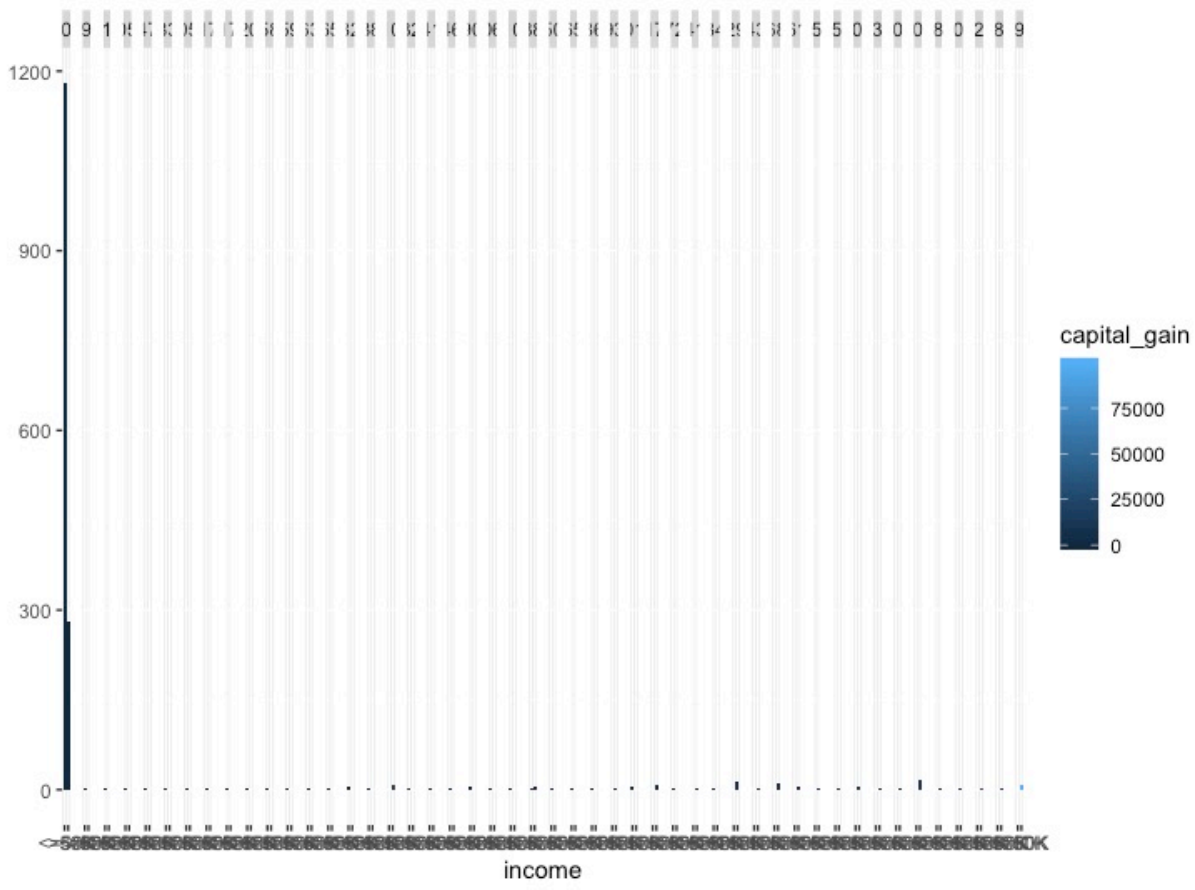
table (train[,c(“capital_gain”, "income")])

This function returns the distribution of income feature against capital_gain.

capital_gain	income <=50K	>50K
0	1180	280
594	3	0
914	1	0
1055	1	0
1471	1	0
1831	1	0
2050	1	0
2174	1	0
2176	3	0
2202	1	0
2580	1	0
2597	1	0
2635	2	0
2653	1	0
2829	5	0
2885	1	0
3103	0	7
3325	1	0
3411	1	0
3464	1	0
3908	4	0
4064	1	0
4101	1	0
4386	1	5
4508	1	0
4650	3	0
4865	2	0
4934	0	1
5013	5	0
5178	0	8
5721	1	0
6418	0	1

6849	2	0
7298	0	13
7430	0	2
7688	0	11
8614	0	4
10520	0	2
13550	0	3
14084	0	4
14344	0	1
15020	0	1
15024	0	17
15831	0	1
20051	0	1
25236	0	1
27828	0	2
99999	0	7

qplot (income, data = train, fill = capital_gain) + facet_grid (. ~ capital_gain)



4. Analysis of capital_loss against Income

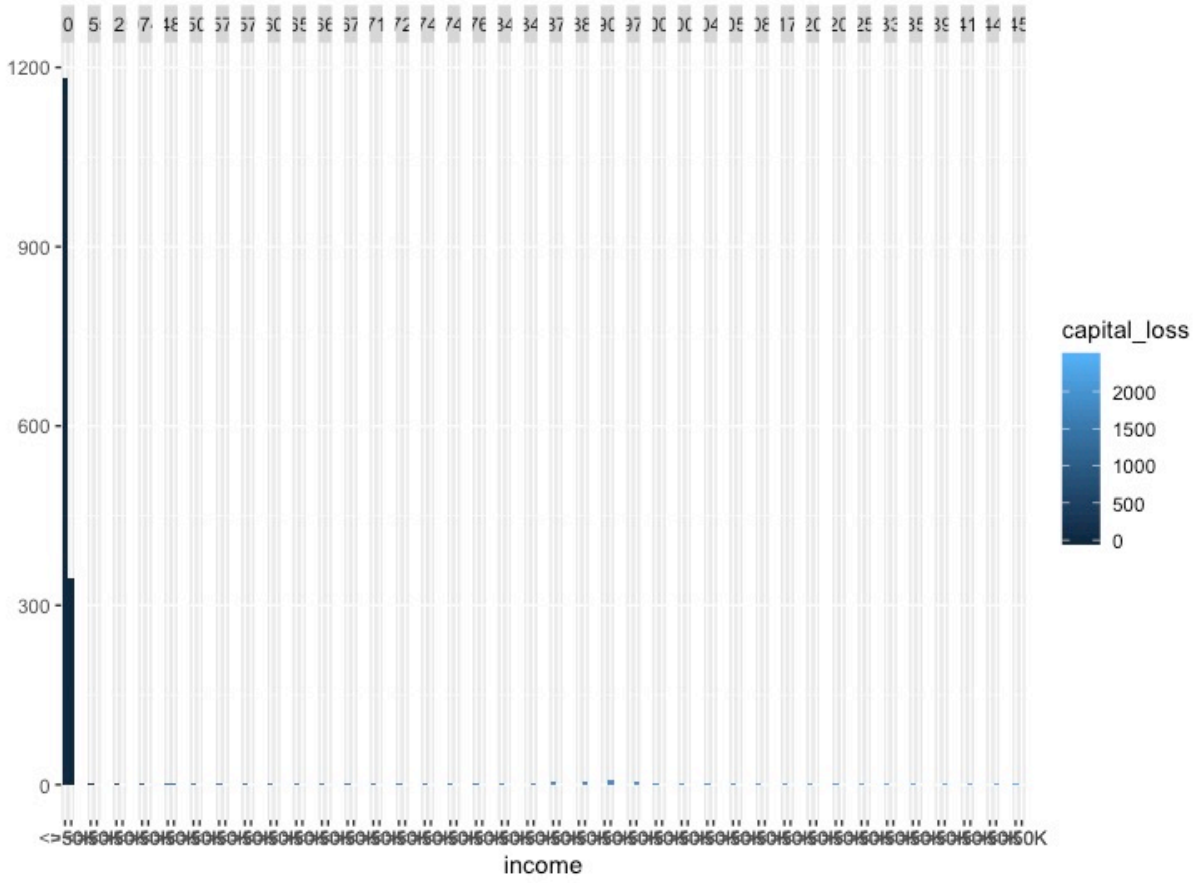
table (train[,c(“capital_loss”, "income")])

This function returns the distribution of income feature against capital_loss.

	income	
capital_loss	<=50K	>50K
0	1182	346
155	1	0
625	1	0
974	1	0
1485	3	1
1504	2	0
1573	1	0
1579	2	0
1602	1	0
1651	2	0
1669	1	0
1672	1	0
1719	2	0
1721	1	0
1740	2	0
1741	2	0
1762	1	0
1844	1	0
1848	0	1
1876	6	0
1887	0	6
1902	0	8
1977	0	6
2001	1	0
2002	2	0
2042	2	0
2051	1	0
2080	1	0
2179	2	0
2205	1	0
2206	1	0
2258	1	0
2339	1	0

2352	1	0
2392	0	1
2415	0	2
2444	0	1
2457	1	0

qplot (income, data = train, fill = capital_loss) + facet_grid (. ~ capital_loss)



5. Analysis of Hours per week against Income

table (train[,c(“hours_per_week”, "income")])

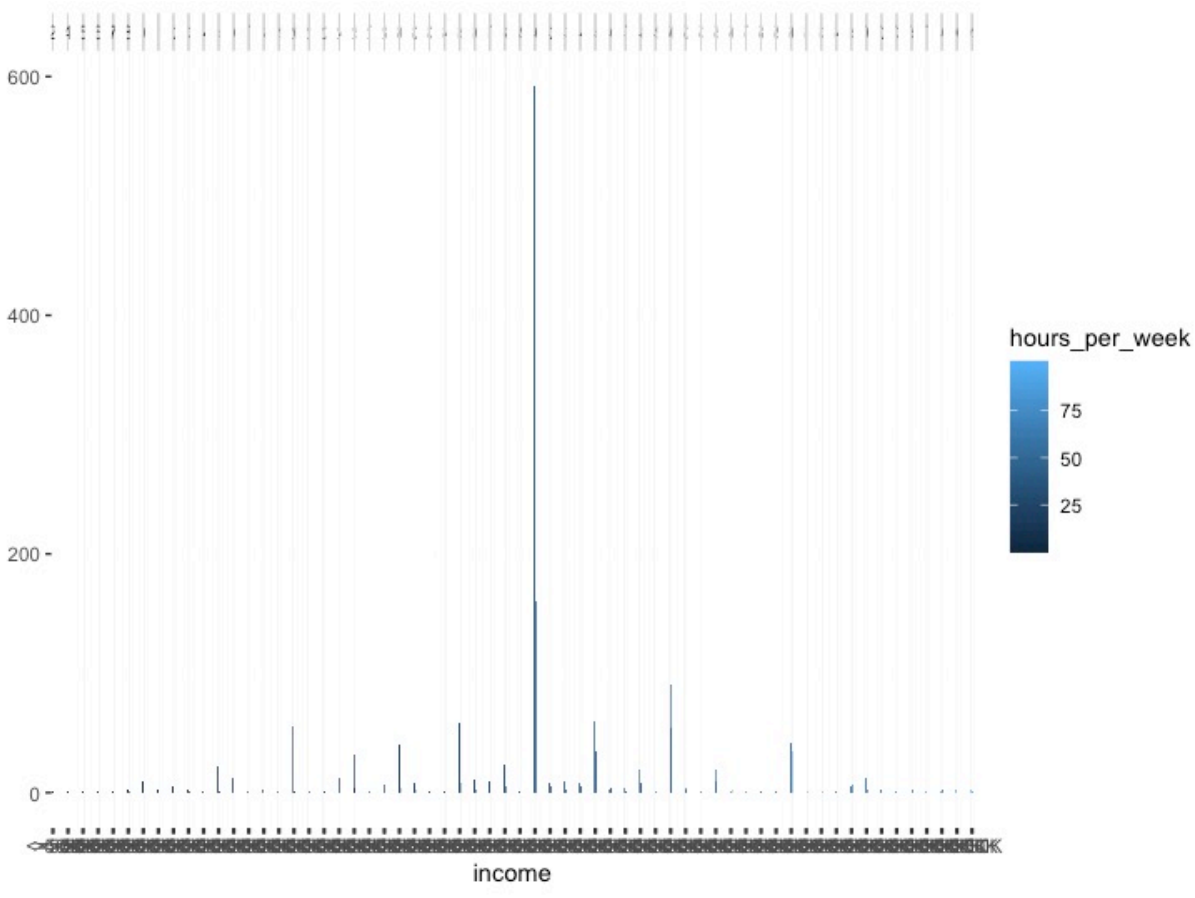
This function returns the distribution of income feature against hours per week.

hours_per_week	income <=50K	>50K
2	1	0

4	1	0
5	1	0
6	1	0
7	1	0
8	3	1
10	9	0
11	2	0
12	6	0
13	2	1
14	1	0
15	22	1
16	13	0
17	1	0
18	2	0
19	1	0
20	56	1
21	1	0
22	1	1
24	13	0
25	32	4
27	1	0
28	7	0
30	40	4
32	8	2
33	1	0
34	1	0
35	59	8
36	11	2
37	9	0
38	24	5
39	1	0
40	592	160
42	8	5
43	10	3
44	8	6
45	60	34
46	2	4
47	4	1
48	20	8
49	1	0

50	91	54
52	4	3
53	1	0
55	20	10
56	1	2
57	1	0
58	1	0
59	1	0
60	42	35
61	0	1
63	0	1
64	1	0
65	5	7
70	12	3
72	3	0
73	1	0
75	0	2
77	1	0
80	1	2
90	2	0
99	3	1

qplot (income, data = train, fill = hours_per_week) + facet_grid (. ~ hours_per_week)



Let’s find the correlation between all the continuous variables.

```
correlation = cor (train[, c("age", "education_num", "capital_gain", "capital_loss",
"hours_per_week")])
diag (correlation) = 0 #Remove self correlations
correlation
```

The following results show that there is no relation between these features and are independent of each other.

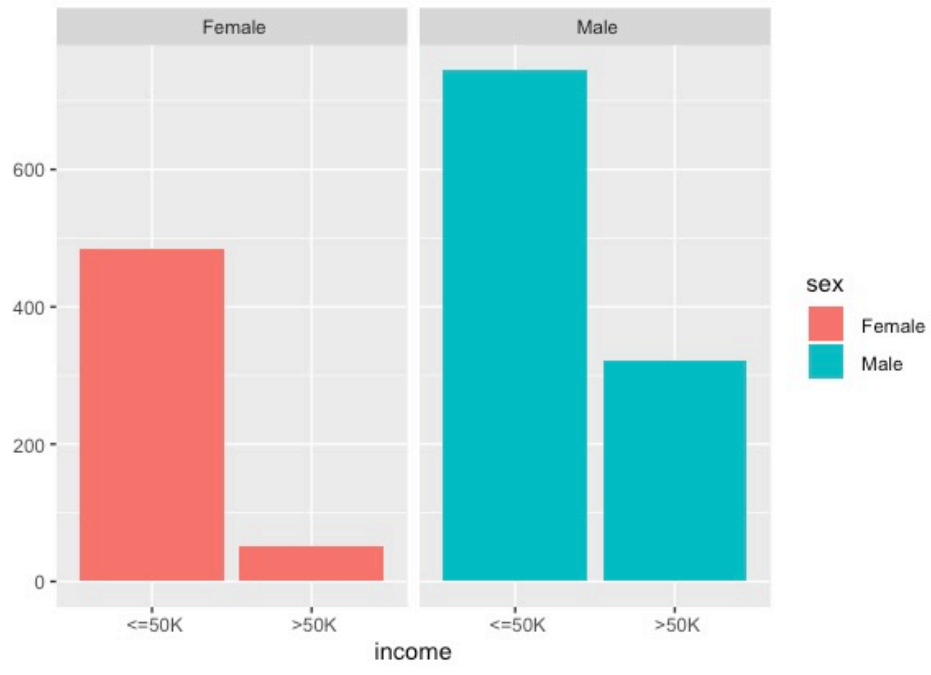
	age	education_num	capital_gain	capital_loss	hours_per_week
age	0	0.003512186	0.06914718	0.04403061	0.13529299
education_num	0.003512186	0	0.14616301	0.06890782	0.1752009
capital_gain	0.069147178	0.14616301	0	-0.03214239	0.09704761
capital_loss	0.044030607	0.068907822	-0.03214239	0	0.04565391
hours_per_week	0.135292986	0.1752009	0.09704761	0.04565391	0

Categorical Variables:

1. Analysis of sex against Income

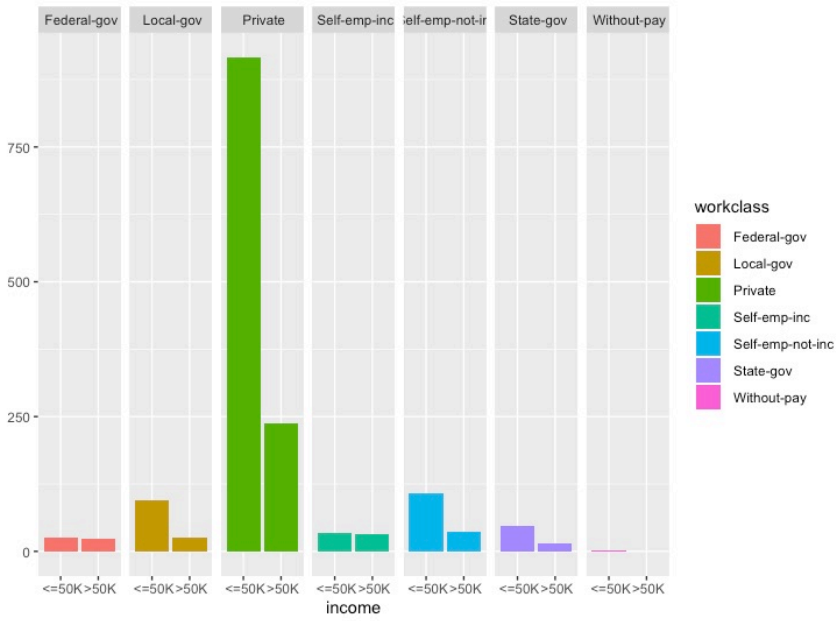
```
                income
sex        <=50K  >50K
  Female    484    51
  Male      744   321
```

```
qplot (income, data = train, fill
= sex) + facet_grid (. ~ sex)
```



2. Analysis of workclass against Income

	income	
workclass	<=50K	>50K
Federal-gov	26	24
Local-gov	95	26
Never-worked	0	0
Private	916	238
Self-emp-inc	35	33
Self-emp-not-inc	107	36
State-gov	48	15
Without-pay	1	0

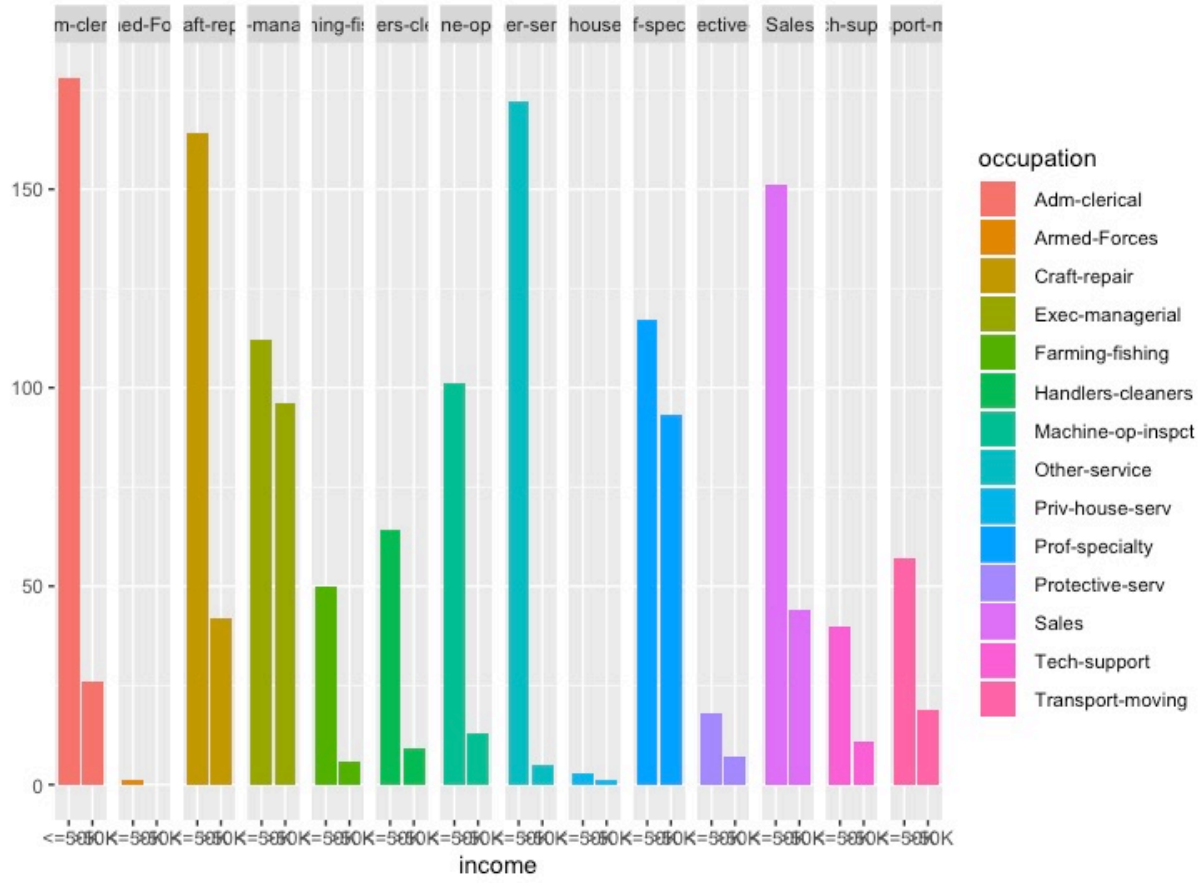


qplot (income, data = train, fill = workclass) + facet_grid (. ~ workclass)

3. Analysis of Occupation against income

	income	
occupation	<=50K	>50K
Adm-clerical	178	26
Armed-Forces	1	0
Craft-repair	164	42
Exec-managerial	112	96
Farming-fishing	50	6
Handlers-cleaners	64	9
Machine-op-inspct	101	13
Other-service	172	5
Priv-house-serv	3	1
Prof-specialty	117	93
Protective-serv	18	7
Sales	151	44
Tech-support	40	11
Transport-moving	57	19

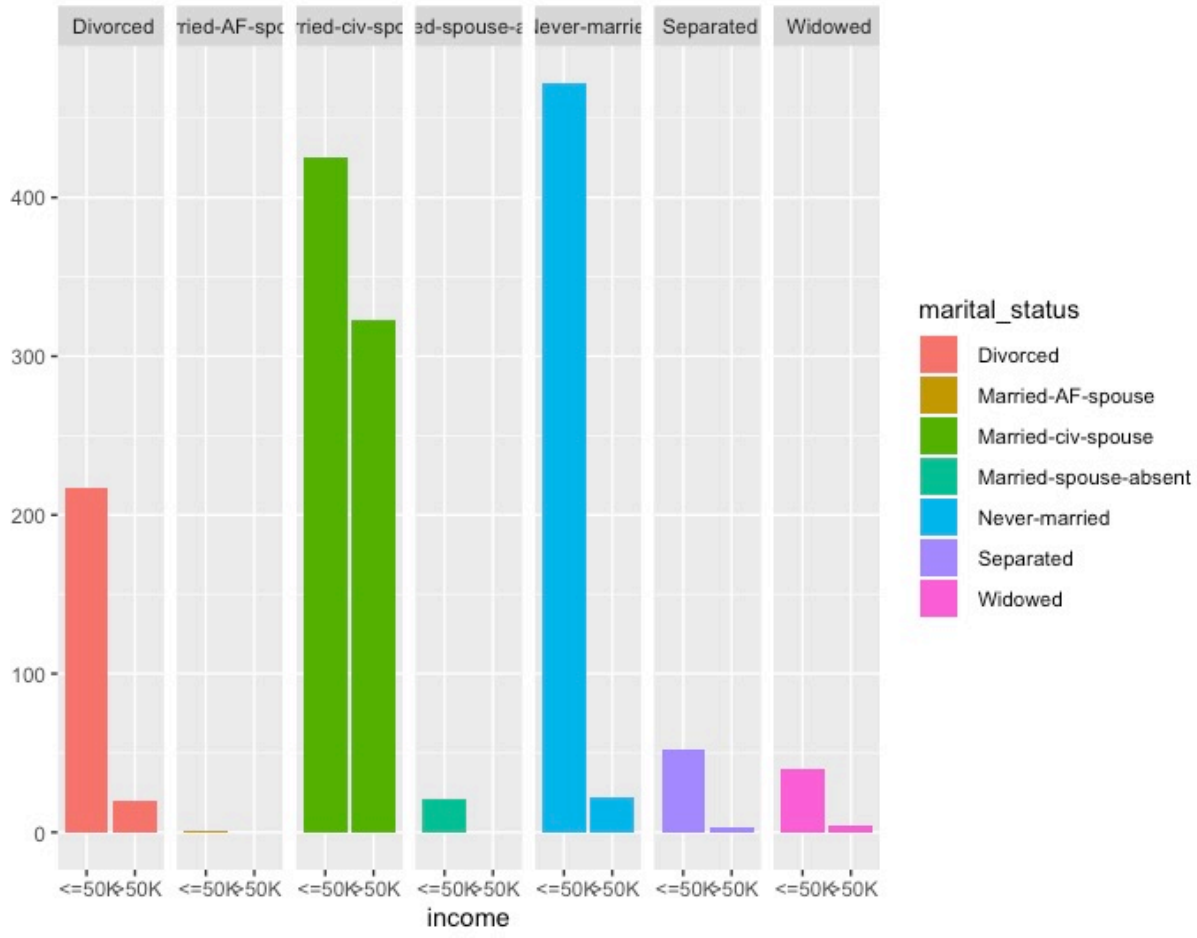
qplot (income, data = train, fill = occupation) + facet_grid (. ~ occupation)



3. Analysis of Marital status against income

	income	
marital_status	<=50K	>50K
Divorced	217	20
Married-AF-spouse	1	0
Married-civ-spouse	425	323
Married-spouse-absent	21	0
Never-married	472	22
Separated	52	3
Widowed	40	4

qplot (income, data = train, fill =marital_status) + facet_grid (. ~ marital_status)

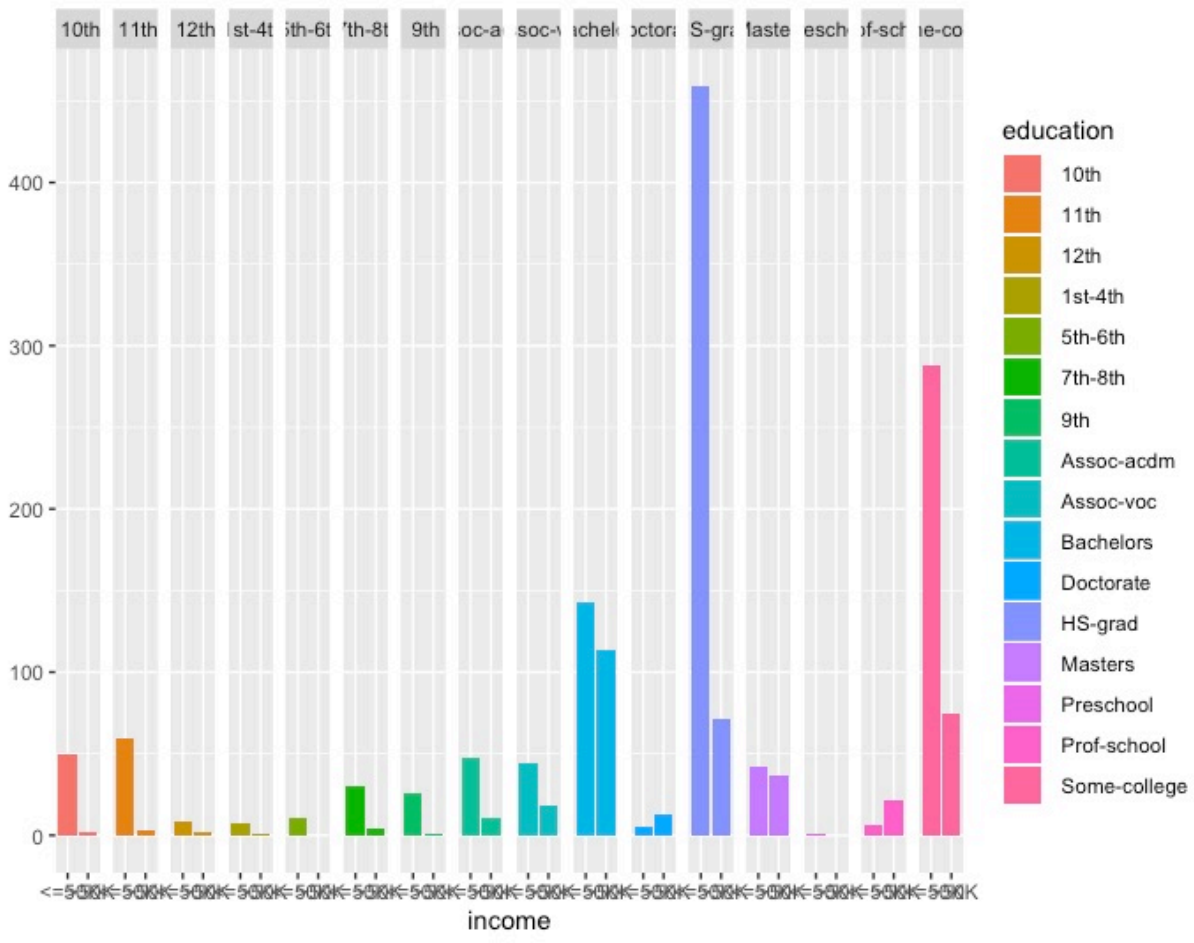


4. Analysis of education against income

	income	
education	<=50K	>50K
10th	50	2
11th	59	3
12th	9	2
1st-4th	7	1
5th-6th	11	0

7th-8th	30	4
9th	26	1
Assoc-acdm	48	11
Assoc-voc	44	18
Bachelors	143	113
Doctorate	5	13
HS-grad	459	71
Masters	42	37
Preschool	1	0
Prof-school	6	21
Some-college	288	75

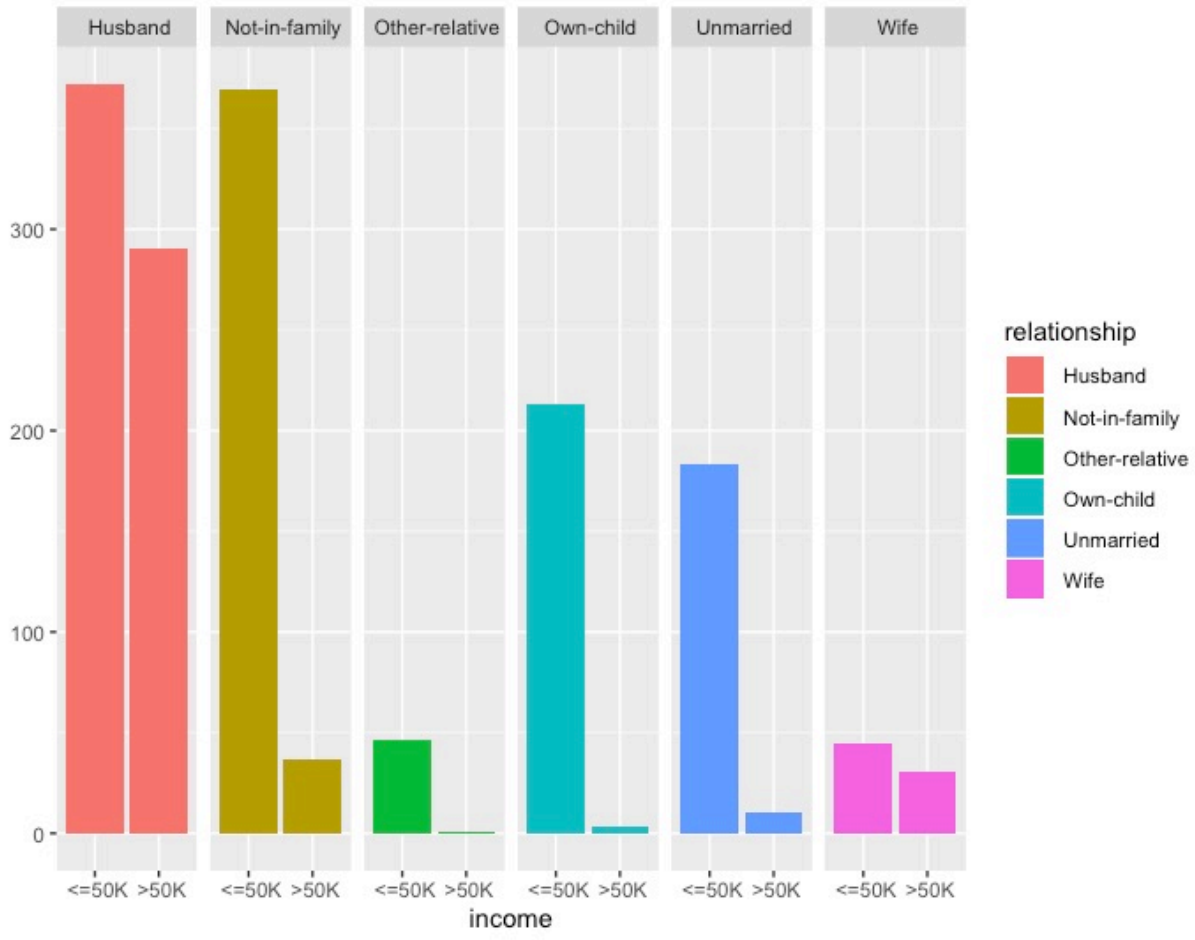
qplot (income, data = train, fill =education) + facet_grid (. ~ education)



5. Analysis of relationship against income

	income	
relationship	<=50K	>50K
Husband	372	290
Not-in-family	369	37
Other-relative	46	1
Own-child	213	3
Unmarried	183	10
Wife	45	31

qplot (income, data = train, fill =relationship) + facet_grid (. ~ relationship)



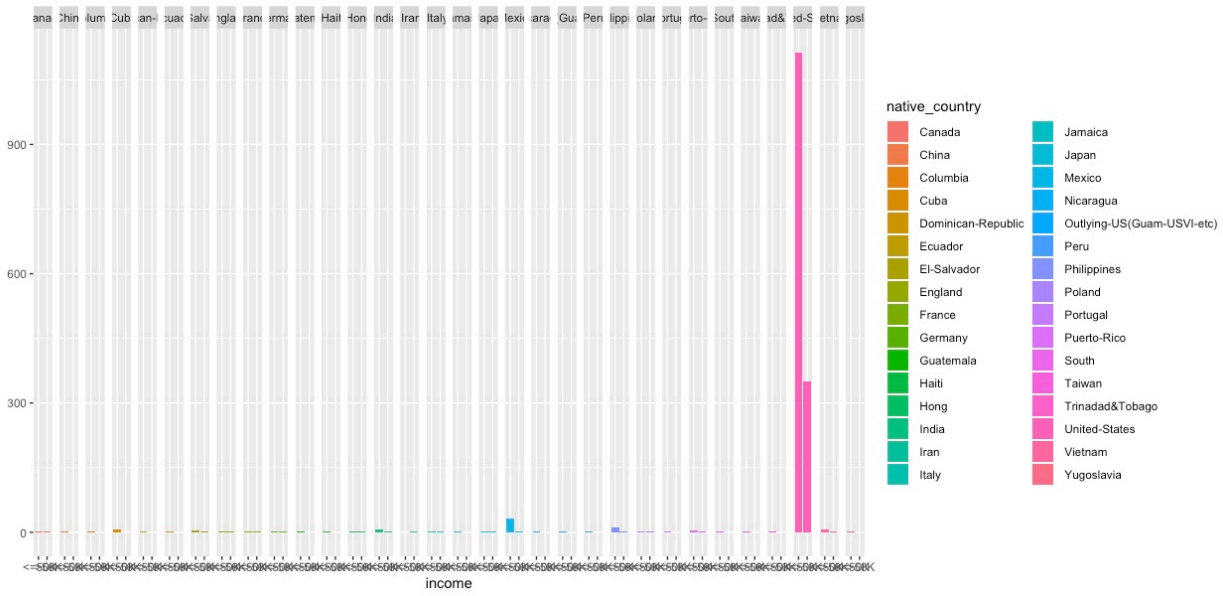
6. Analysis of native country against income

	income	
native_country	<=50K	>50K

Cambodia	0	0
Canada	3	3
China	2	0
Columbia	3	0
Cuba	7	0
Dominican-Republic	3	0
Ecuador	2	0
El-Salvador	5	1
England	1	2
France	1	1
Germany	3	1
Greece	0	0
Guatemala	1	0
Haiti	3	0
Holand-Netherlands	0	0
Honduras	0	0
Hong	1	1
Hungary	0	0
India	6	1
Iran	0	1
Ireland	0	0
Italy	2	1
Jamaica	3	0
Japan	1	2
Laos	0	0
Mexico	32	3
Nicaragua	3	0
Outlying-US(Guam-USVI-etc)	1	0
Peru	2	0
Philippines	11	3
Poland	2	1
Portugal	1	0
Puerto-Rico	5	1
Scotland	0	0
South	1	0
Taiwan	2	0
Thailand	0	0
Trinadad&Tobago	1	0
United-States	1113	349
Vietnam	6	1

Yugoslavia 1 0

qplot (income, data = train, fill = native_country) + facet_grid (. ~ native_country)



According to the analysis made above,

- On finding out the summary of age, it is observed that age has got a wide range and variability. The distribution of mean for the income levels <=50k and >50k show a huge difference, implying that data is more uncertain/impure. This indicates that age can be a good predictor.
- On summarizing education_num, hours_per_week, capital_gain and capital_loss, it is again observed that all these features have a wide range of distribution in their means, turning out to be good predictors of income level.
- Coming to categorical variables, the sex feature is eliminated or withheld since it does not give much information on the income levels of an individual. It is not so sparse enough to give more knowledge on whether being a male/female depends on how much an individual earns. Also, the attributes - race and fnlwgt do not contribute much in the prediction of income.
- The features workclass, marital_status, education, occupation, relationship all provide enough information to make a prediction. Since the goal is to predict the income in USD, native country is also taken into account as it plays a role in deciding which group of people, belonging to a particular place, earn the required income.

Step 2: Cleaning the dataset

The dataset includes many rows/tuples with missing values that are represented with a ?. These values need to be removed for the correct analysis of the data and for correct prediction. The following lines show the cleaning of dataset in R.

```
data<-read.delim("/Users/navyasogi/Downloads/census-adult.txt",
                         sep = ",",
                            header = FALSE,
                           na.strings = " ?")

data<-na.omit(data)
row.names(data)<-1:nrow(data)
View(data)
```

The original dataset now reduces to 30,162 rows.

Step 3: Obtaining Sample Data and Splitting into Train/Test Data

In this step, we obtain a sample of 2000 rows from the original dataset using the following:

```
set.seed(40)
sample_data<- sample_n(data,2000)
View(sample_data)
```

Seed value is used as a random number generator that produces the same set of sample data every time it is run. Next, we split the sample_data into two sets - train, on which the analysis is done and a model is built & test, that behaves like a validation model against the train dataset. The splitting is done as follows:

```
sample <- sample.int(n = nrow(sample_data), size = floor(.8*nrow(sample_data)),
replace = F)
train <- sample_data[sample, ]
test  <- sample_data[-sample, ]
View(train)
View(test)
```

The split ratio given is 80/20. After splitting the sample_data into train and test, we have 1600 rows in train and 400 rows in test.

Step 4: Analyzing the Train Dataset using Decision Tree with Information Gain and Feature Selection

For our analysis, we have used decision tree classifier with information gain. A decision tree classifies a dataset based on two conditions: Yes Or No.
Our target is to predict whether an individual earns more than 50,000 USD or less than or equal to 50,000 USD. To do this, we compute the entropy and information gain for every attribute.
Entropy is a measure of how impure the data is. It refers to the homogeneity of the dataset. If the dataset is highly homogenous, then the entropy is 0. This means we have complete knowledge about the data. On the other hand, if the dataset is widely distributed, the entropy increases. Information gain measures how much information a feature gives us about the dataset.
Formula for entropy and information gain are as follows:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Where,
pi is the probability of an element i in our dataset

$$IG(Y, X) = E(Y) - E(Y|X)$$

Where,
Y is the target feature to be predicted and X is a feature of dataset that is analyzed against the target. Information Gain of a particular feature will be the weighted entropy of the target(Y) minus the entropy of Y given the entropy of the particular feature.

We calculate entropy and information gain as follows:

```
entropy <- function(target) {
  freq <- table(target)/length(target)
  # vectorize
  vec <- as.data.frame(freq)[,2]
  #drop 0 to avoid NaN resulting from log2
  vec<-vec[vec>0]
```

```
  #compute entropy
  -sum(vec * log2(vec))
}

IG_cal<-function(data,target,feature){
  #Strip out rows where feature is NA
  data<-data[!is.na(data[,feature]),]
  #use dplyr to compute e and p for each value of the feature
  dd_data <- data %>% group_by_at(feature) %>% summarise(e=entropy(get(target)),
                                                         n=length(get(target))
  )

  #compute entropy for the parent
  e0<-entropy(data[,target])
  #calculate p for each value of feature
  dd_data$p<-dd_data$n/nrow(data)
  #compute IG
  IG<-e0-sum(dd_data$p*dd_data$e)

  return(IG)
}
```

A decision tree decides the node on which it has to split based on the feature that has the highest information gain. Higher the impurity of a feature, more is the information obtained from it.

Information Gain of individual feature		**Entropy of individual feature**	
IG_cal(train, “age”, “income”)	0.1102322	entropy(train$age)	5.570075
IG_cal(train, “marital_status”,”income”)	0.1556029	entropy(train$marital_status)	1.842837
IG_cal(train, “workclass”,”income”)	0.01838583	entropy(train$workclass)	1.473397
IG_cal(train, “occupation”,”income”)	0.08820544	entropy(train$occupation)	3.381529

Information Gain of individual feature		Entropy of individual feature	
IG_cal(train, “hours_per_week”, “income”)	0.07675161	entropy(train$hours_per_week)	3.37544
IG_cal(train, “relationship”, “income”)	**0.161055**	entropy(train$relationship)	2.145213
IG_cal(train, “education_num”,”income”)	0.09712848	entropy(train$education_num)	2.883837
IG_cal(train, “education”,”income”	0.09712848	entropy(train$education)	2.883837
IG_cal(train, “capital_gain”,”income”	0.1364278	entropy(train$capital_gain)	0.8564457
IG_cal(train,”capital_loss”,”income”	0.04329143	entropy(train$capital_loss)	0.4808308
IG_cal(train,”native_country”,”income”	0.01682794	entropy(train$native_country)	0.7862466

Based on the above information, feature relationship has the highest information gain. This means that the relationship feature provides more knowledge, less homogenous and the data is widely distributed about its mean. Hence, the decision tree chooses **relationship** as the best split and it is the root node of the tree.

Step 5: Building the decision tree

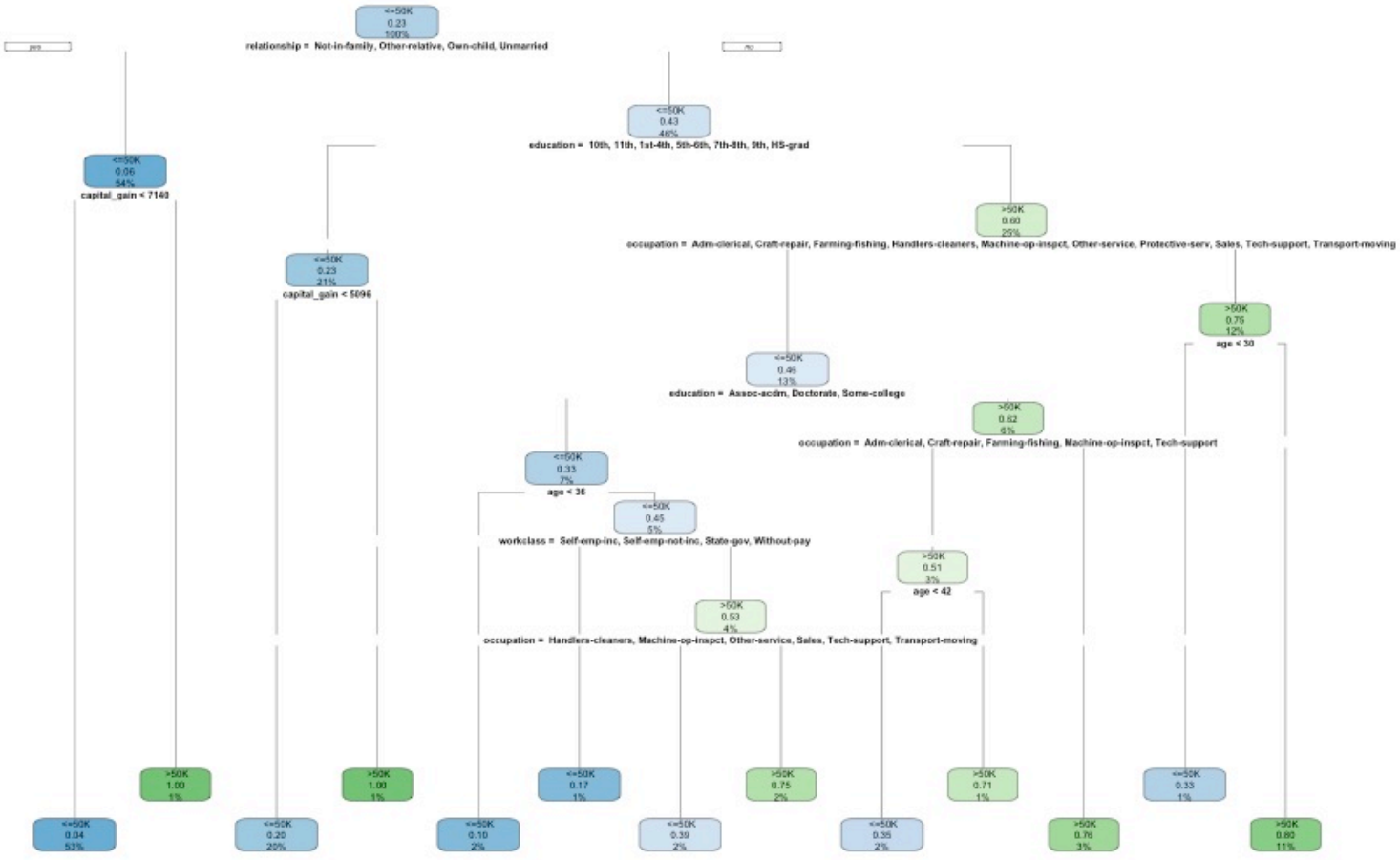
1. First, compute the entropy for the entire dataset
2. Next, for every attribute:
    1. calculate entropy for all categorical features
    2. take the average information entropy for the current feature
    3. calculate information gain for the current feature
3. Pick the feature that has the highest gain attribute.
4. Repeat recursively until we get the tree we desired.

The tree is plotted as follows:

install.packages("rpart")

```
install.packages("rpart.plot")
library(rpart)
library(rpart.plot)
fit <- rpart(income ~ age + education + education_num + marital_status + occupation +
capital_gain + capital_loss + native_country + workclass  + hours_per_week +
relationship, data=train)
rpart.plot(fit)
```

After withholding the features sex, fnwgt, race according to the above analysis, the resulting decision tree is:



Files Attached:

A PDF document that explains the coding in R
A .R file for code execution