Data Science

Data Cleansing

Data Cleansing

When dealing with real world data, you have to keep in mind that it is extremely untidy. It will not have a proper structure and hence this is where data cleansing comes in to bring proper structure to this data.

Data Cleansing

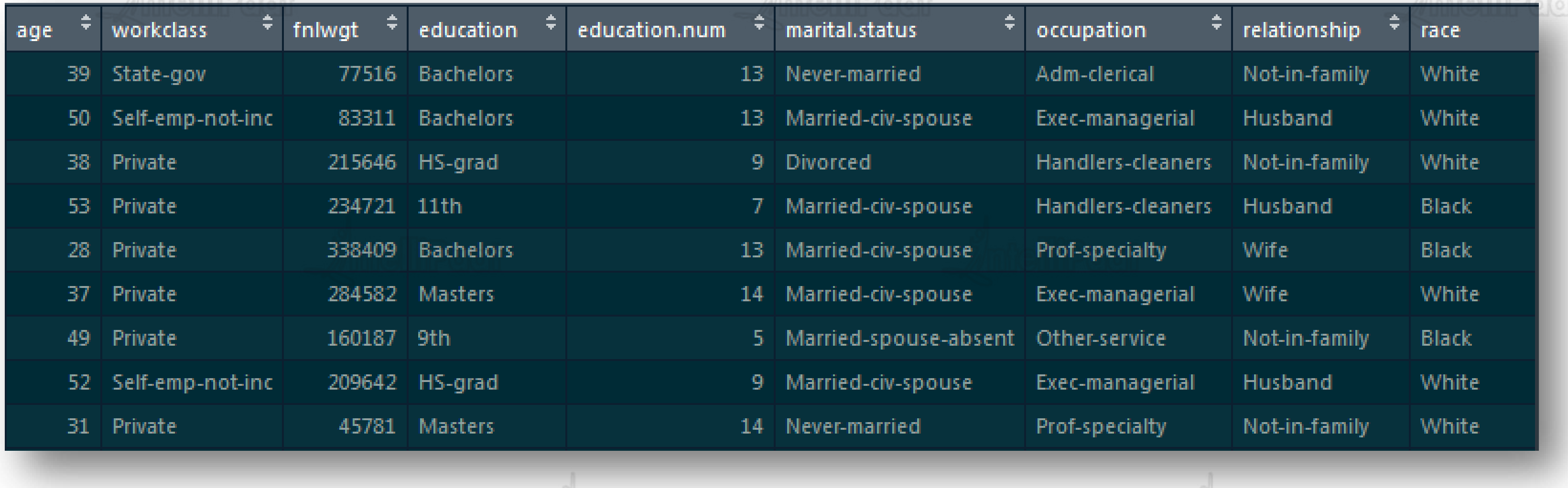
These are some of the actions which you'd have to take during data cleansing process:

- Giving proper names to columns
- Checking for whitespaces in data
- Grouping of similar data into same levels
- Handling missing values(Imputation)

Data Cleansing on ‘Census’ Data

Census Data

This census data has 32561 rows and 15 columns

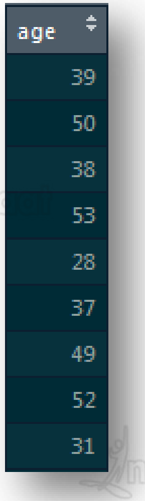


age	workclass	fnlwgt	education	education.num	marital.status	occupation	relationship	race
39	State-gov	77516	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White
50	Self-emp-not-inc	83311	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White
38	Private	215646	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White
53	Private	234721	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black
28	Private	338409	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black
37	Private	284582	Masters	14	Married-civ-spouse	Exec-managerial	Wife	White
49	Private	160187	9th	5	Married-spouse-absent	Other-service	Not-in-family	Black
52	Self-emp-not-inc	209642	HS-grad	9	Married-civ-spouse	Exec-managerial	Husband	White
31	Private	45781	Masters	14	Never-married	Prof-specialty	Not-in-family	White

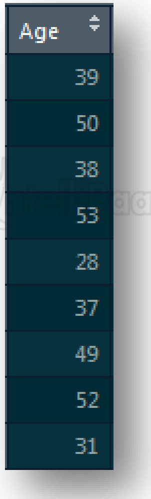
Data Cleansing Steps – Renaming Columns

Renaming the age column:

```
colnames(census)[colnames(census)=="age"]<-"Age"
```



age
39
50
38
53
28
37
49
52
31

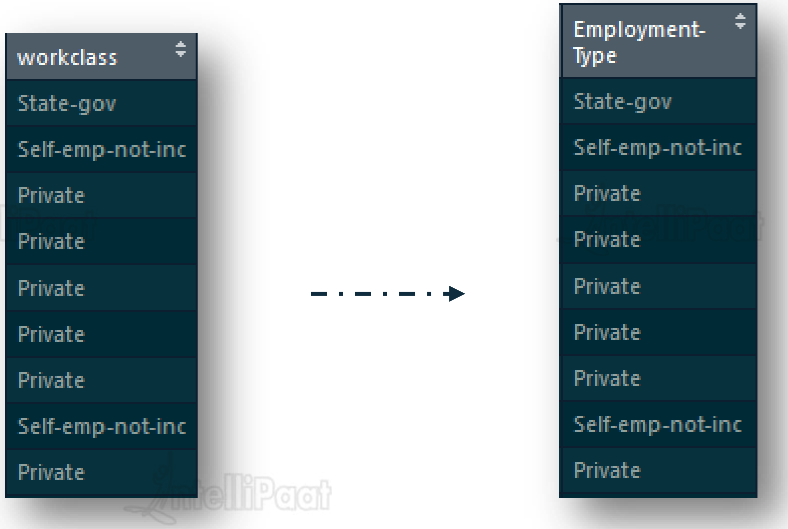


Age
39
50
38
53
28
37
49
52
31

Data Cleansing Steps – Renaming Columns

Renaming ‘workclass’ column:

```
library(data.table)
setnames(census, "workclass", "Employment-Type")
```



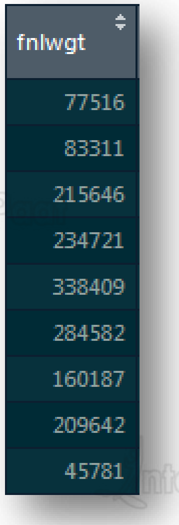
workclass
State-gov
Self-emp-not-inc
Private
Private
Private
Private
Private
Self-emp-not-inc
Private

Employment-Type
State-gov
Self-emp-not-inc
Private
Private
Private
Private
Private
Self-emp-not-inc
Private

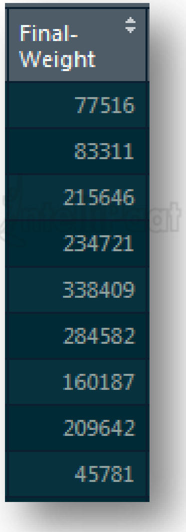
Data Cleansing Steps – Renaming Columns

Renaming ‘fnlwgt’ column:

```
library(plyr)
census<-rename(census,c('fnlwgt'='Final-Weight'))
```



fnlwgt
77516
83311
215646
234721
338409
284582
160187
209642
45781

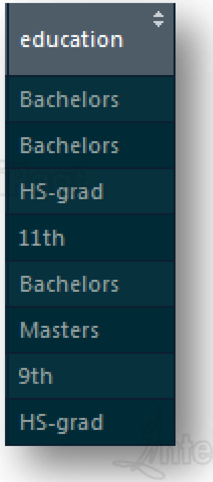


Final-Weight
77516
83311
215646
234721
338409
284582
160187
209642
45781

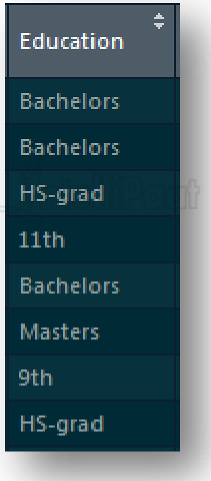
Data Cleansing Steps – Renaming Columns

Renaming ‘education’ column:

```
library(gdata)
census <- rename.vars(census, from = "education", to = "Education")
```



education
Bachelors
Bachelors
HS-grad
11th
Bachelors
Masters
9th
HS-grad



Education
Bachelors
Bachelors
HS-grad
11th
Bachelors
Masters
9th
HS-grad

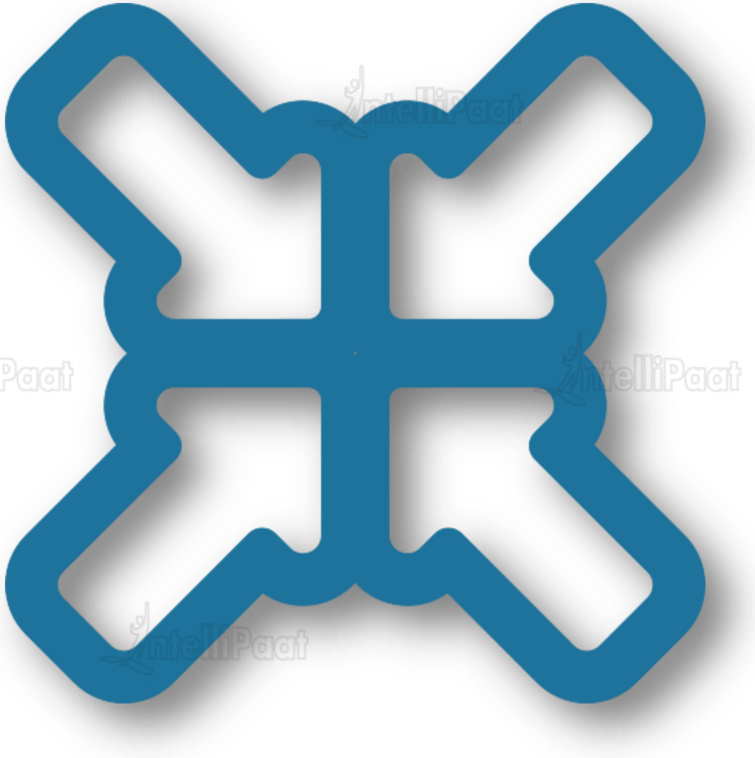
Data Cleansing Steps – Renaming Columns

Renaming rest of the columns:

```
setnames(census, "education.num", "Education-Number")
setnames(census, "marital.status", "Marital-Status")
setnames(census, "occupation", "Occupation")
setnames(census, "relationship", "Relationship")
setnames(census, "race", "Race")
setnames(census, "sex", "Sex")
setnames(census, "capital.gain", "Capital-Gain")
setnames(census, "capital.loss", "Capital-Loss")
setnames(census, "hours.per.week", "Hours-Per-Week")
setnames(census, "native.country", "Native-Country")
setnames(census, "X", "Income")
```

Data Cleansing Steps – Collapsing Levels

Many times a categorical column has levels which represent the same thing. So, in this case, the repetitive levels could be collapsed into a common level. There are also chances where multiple levels which come under the same category can be grouped under an umbrella level



Data Cleansing Steps – Collapsing Levels

- In ‘Employment-Type’ column, collapsing “State-gov”, “Federal-gov” & “Local-gov” into “Government”.
- Also, collapsing ‘Self-emp-inc’ & ‘Self-emp-not-inc’ into “Self Employed”

```
table(census$`Employment-Type`)
as.character(census$`Employment-Type`) -> census$`Employment-Type`
```

```
census$`Employment-Type`[census$`Employment-Type`==" State-gov"] <- "Government"
census$`Employment-Type`[census$`Employment-Type`==" Federal-gov"] <- "Government"
census$`Employment-Type`[census$`Employment-Type`==" Local-gov"] <- "Government"

census$`Employment-Type`[census$`Employment-Type`==" Self-emp-inc"] <- "Self Employed"
census$`Employment-Type`[census$`Employment-Type`==" Self-emp-not-inc"] <- "Self Employed"
```

Data Cleansing Steps – Collapsing Levels

In ‘Marital-Status’ column, collapsing ‘Married-AF-spouse’, ‘Married-spouse-absent’, & ‘Married-civ-spouse’ into “Married”

```
table(census$`Marital-Status`)
as.character(census$`Marital-Status`) -> census$`Marital-Status`
```

```
census$`Marital-Status`[census$`Marital-Status`== " Married-AF-spouse"] <- "Married"
census$`Marital-Status`[census$`Marital-Status`== " Married-spouse-absent"] <- "Married"
census$`Marital-Status`[census$`Marital-Status`== " Married-civ-spouse"] <- "Married“
```

Data Cleansing Steps – Collapsing Levels

In ‘Native-Country’ column, collapsing different levels into “Europe”:

```
table(census$`Native-Country`)
as.character(census$`Native-Country`) -> census$`Native-Country`
```

```
census$`Native-Country`[census$`Native-Country`==" England"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" France"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" Germany"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" Greece"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" Ireland"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" Scotland"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" Portugal"] <- "Europe"
census$`Native-Country`[census$`Native-Country`==" Italy"] <- "Europe"
```

Data Cleansing Steps – Collapsing Levels

In ‘Native-Country’ column, collapsing different levels into “Asia”:

```
census$`Native-Country`[census$`Native-Country`==" India"] <- "Asia"
census$`Native-Country`[census$`Native-Country`==" Vietnam"] <- "Asia"
census$`Native-Country`[census$`Native-Country`==" Taiwan"] <- "Asia"
census$`Native-Country`[census$`Native-Country`==" Japan"] <- "Asia"
census$`Native-Country`[census$`Native-Country`==" Thailand"] <- "Asia"
census$`Native-Country`[census$`Native-Country`==" Iran"] <- "Asia"
census$`Native-Country`[census$`Native-Country`==" China"] <- "Asia"
```

Data Cleansing Steps – Imputation

Wherever we have “ ?”, we’ll replace it with NA:

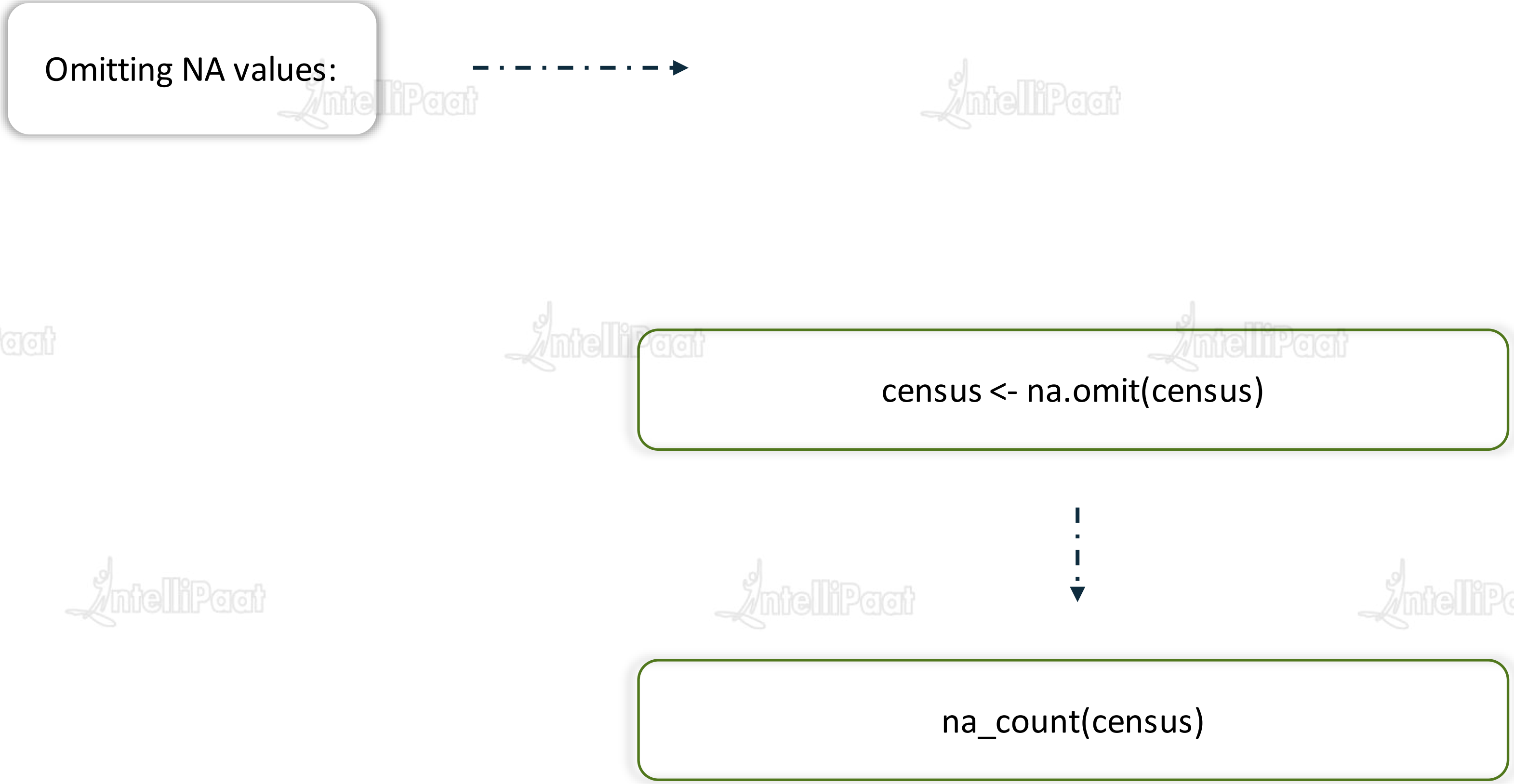
```
census[census == " ?"] <- NA
```

Creating function to count number of NA values:

```
na_count <-function (x) sapply(x, function(y) sum(is.na(y)))
```

```
na_count(census)
```

Data Cleansing Steps – Imputation



Omitting NA values:

```
census <- na.omit(census)
```

```
na_count(census)
```

Thank You