CHULA ENGINEERING
Foundation toward Innovation

COMPUTER

Chula Big Data and IoT
Center of Excellence
(CUBIC)

Data Mining Group
Machine Intelligence and Knowledge Discovery Lab
Chulalongkorn University

Data Preparation (Data Cleansing)

Python for Data Analytics

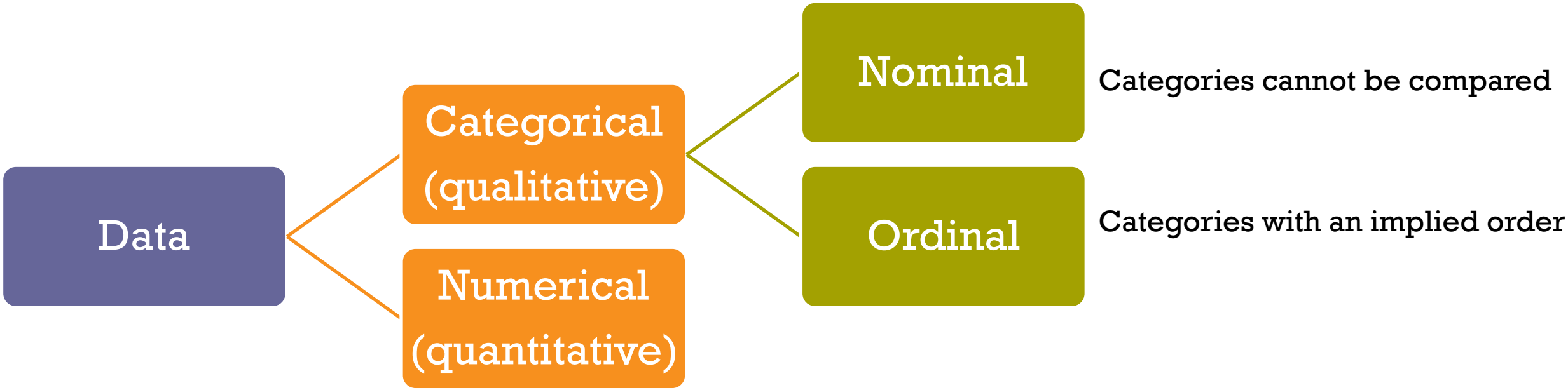
Peerapon Vateekul, Ph.D.
Department of Computer Engineering,
Faculty of Engineering, Chulalongkorn University
Peerapon.v@chula.ac.th

Terminology: Data table

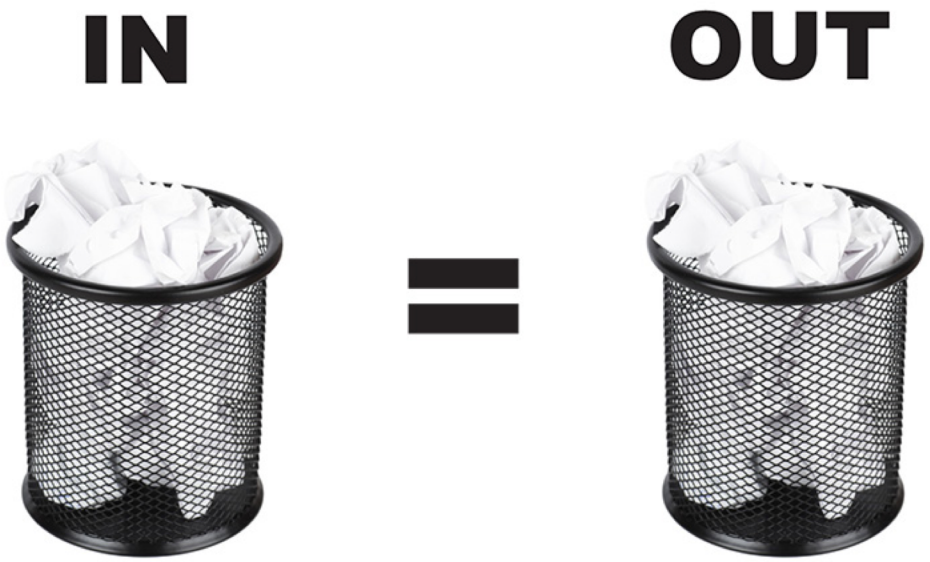
		inputs		target
Age	**Income**	**Gender**	**Province**	**Purchase**
25	25,000	Female	Bangkok	Yes
35	50,000	Female	Nontaburi	Yes
32	35,000	Male	Bangkok	No

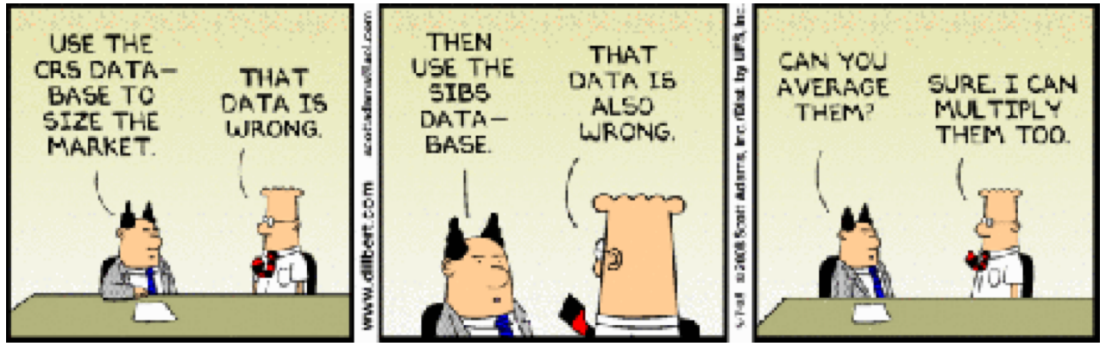
- Row
  - Example, instance, case, observation, subject
- Column
  - Feature, variable, attribute
- Input
  - Predictor, independent, explanatory variable
- Target
  - Output, outcome, response, dependent variable

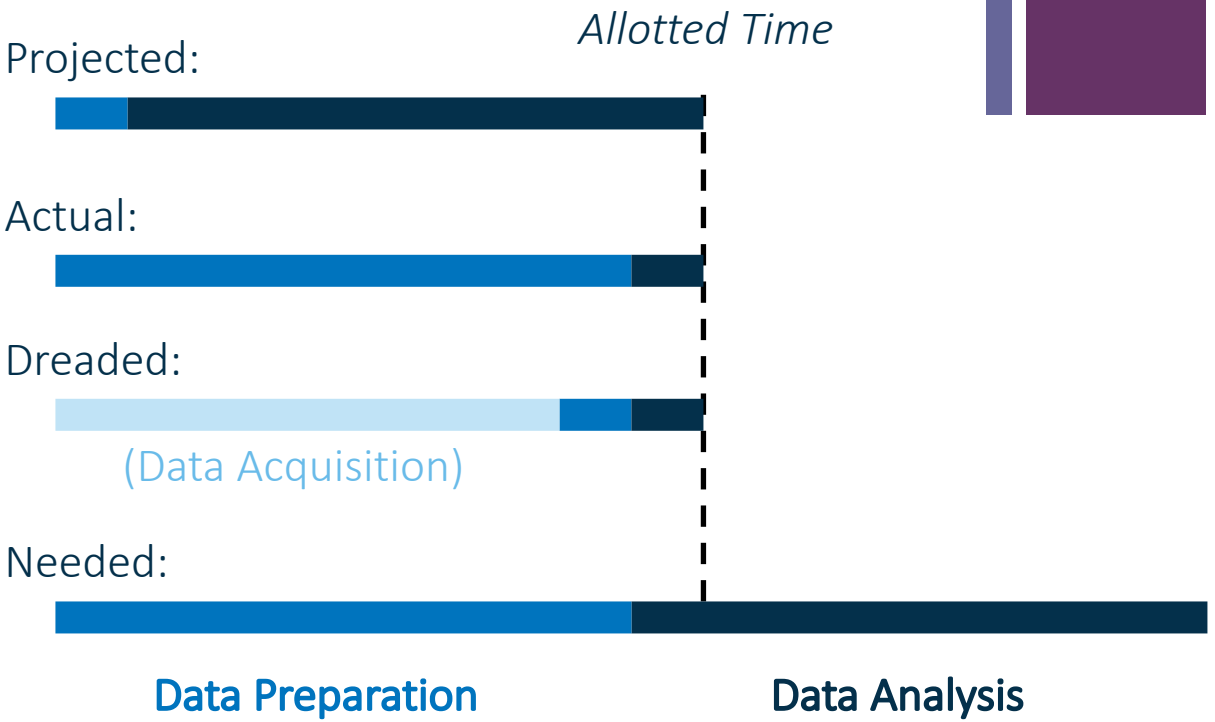
Terminology: Kinds of data



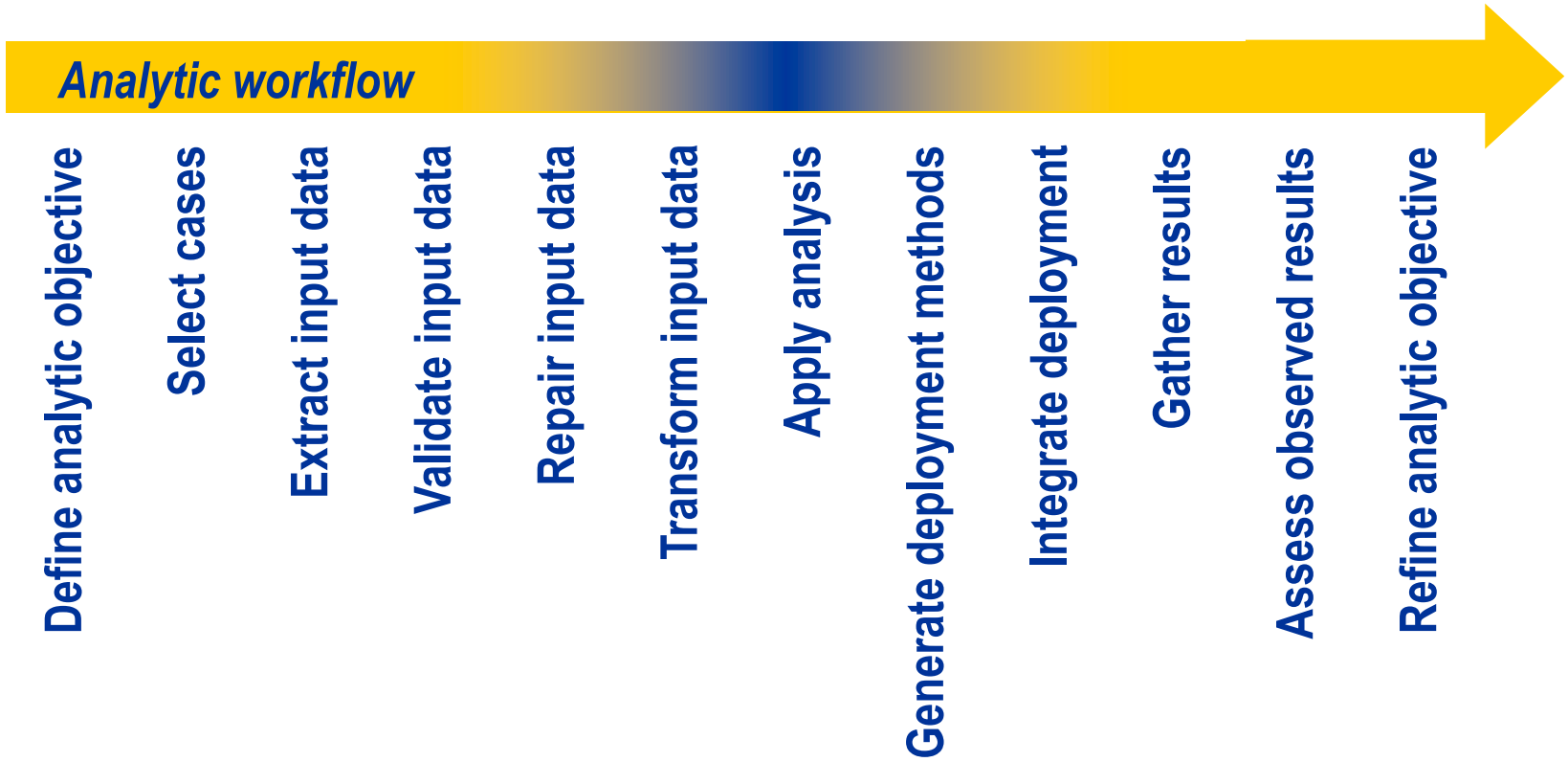
Data preparation is very important!







Analytics workflow



Data preparation challenges

- Massive data sets

- Temporal infidelity

- Transaction and event data

- Non-numeric data

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x_1 + \hat{w}_2 x_2$$

- Exceptional, extreme, and missing values

- Stationarity


Practice is everything.
Periander
quotefancy

DATAQUEST BLOG

28 DECEMBER 2016 / DATA CLEANING

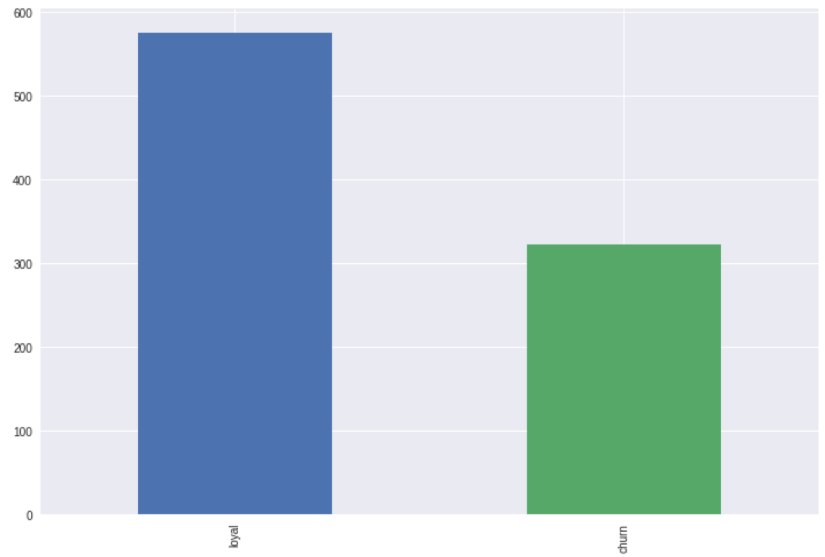
Preparing and Cleaning Data for Machine Learning

- **LoadAndPreviewData.ipynb**
- 1) Load & preview Data
  - Histogram & missing
- **DataPreparation.ipynb**
- 2) Drop unrelated variables
- 3) Drop records with missing target
- 4) Drop single value inputs
- 5) Impute missing values
  - Numeric variable (mean)
  - Categorical variable (mode)
- 6) Convert non-numeric to numeric variable
  - Ordinal variable
  - Nominal variable
- 7) Create calculated variables
- 8) Remove outlier
- 9) Save data

https://www.dataquest.io/blog/machine-learning-preparing-data/

1) Load & Preview Data

- Data set "customer-churn-data_for-data-prep.csv"
- Target "Churn" or "Royal"




```
[ ]  df = pd.read_csv('customer-churn-data_for-data-prep.csv')
```

```
[ ]  df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 996 entries, 0 to 995
Data columns (total 7 columns):
Gender              978 non-null object
Payment Method      996 non-null object
Churn               898 non-null object
LastTransaction     996 non-null int64
Country             996 non-null object
Province            427 non-null object
BirthYear           982 non-null float64
dtypes: float64(1), int64(1), object(5)
memory usage: 54.5+ KB
```

	Gender	Payment Method	Churn	LastTransaction	Country	Province	BirthYear
0	male	credit card	loyal	1	TH	Bangkok	1899.0
1	female	credit card	churn	3	TH	Bangkok	2000.0
2	male	credit card	loyal	6	TH	Non Nam Thaeng	1980.0
3	female	credit card	churn	7	TH	Ang Thong	1899.0
4	female	credit card	churn	11	TH	Bueng Kan	1985.0

2) Drop unrelated variables

```
df = df.drop(['LastTransaction'], axis=1)
```

	Gender	Payment Method	Churn	LastTransaction	Country	Province	BirthYear
0	male	credit card	loyal	1	TH	Bangkok	1899.0
1	female	credit card	churn	3	TH	Bangkok	2000.0
2	male	credit card	loyal	6	TH	Non Nam Thaeng	1980.0
3	female	credit card	churn	7	TH	Ang Thong	1899.0
4	female	credit card	churn	11	TH	Bueng Kan	1985.0

Before

	Gender	Payment Method	Churn	Country	Province	BirthYear
0	male	credit card	loyal	TH	Bangkok	1899.0
1	female	credit card	churn	TH	Bangkok	2000.0
2	male	credit card	loyal	TH	Non Nam Thaeng	1980.0
3	female	credit card	churn	TH	Ang Thong	1899.0
4	female	credit card	churn	TH	Bueng Kan	1985.0

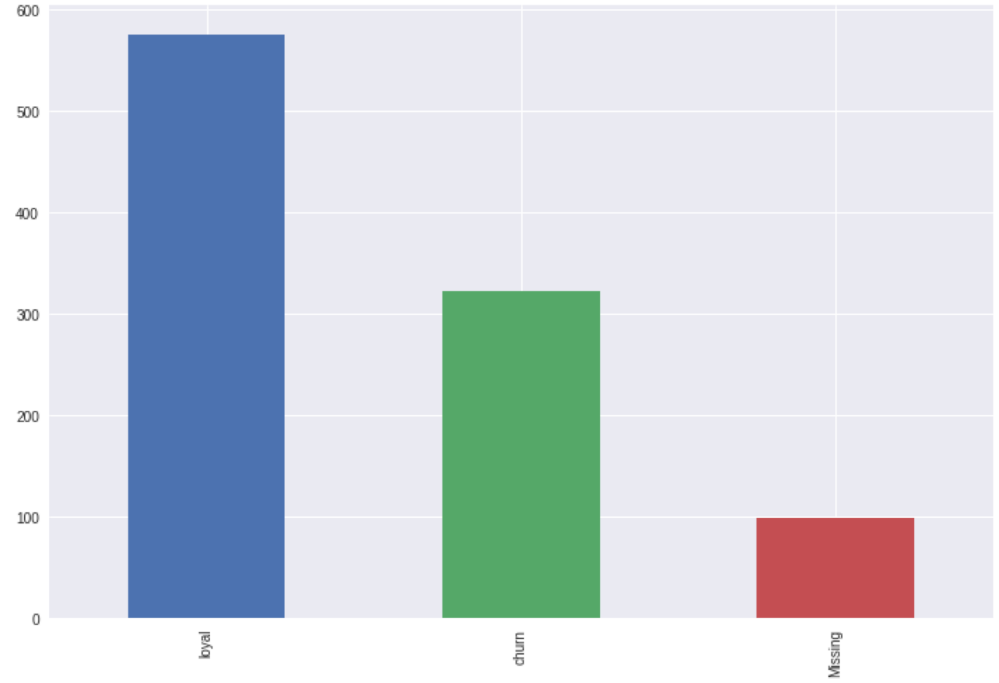
After

3) Drop records with missing target

```
df = df.dropna(axis=0, subset=['Churn'])
```

```
[ ] df['Churn'].fillna('Missing').value_counts().plot(kind='bar')
```

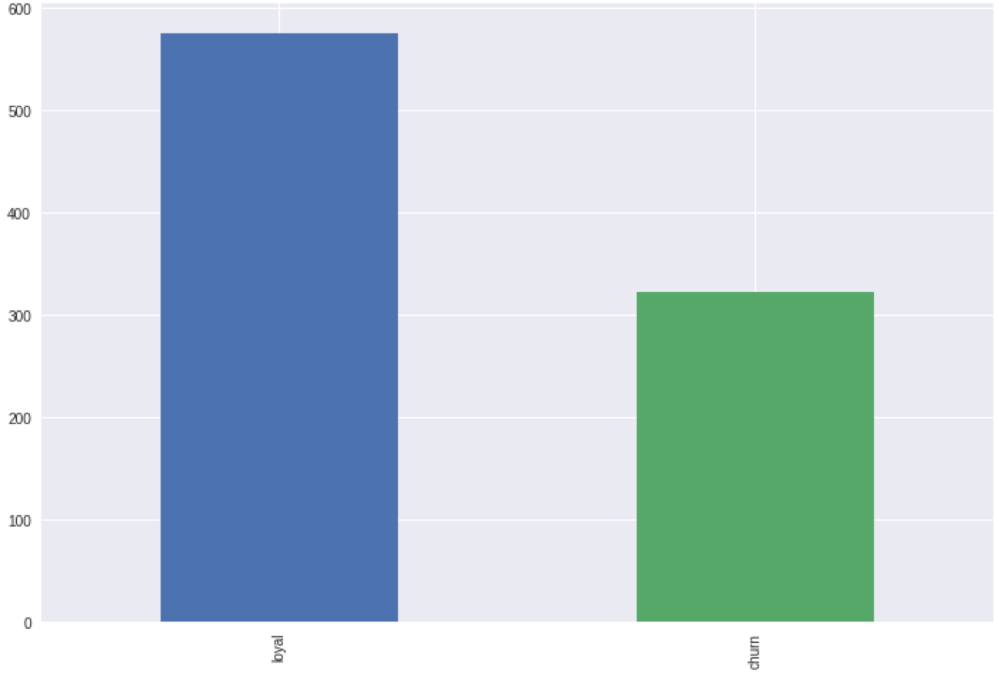
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f73cccc7490>
```



Before

```
[ ] df['Churn'].fillna('Missing').value_counts().plot(kind='bar')
```

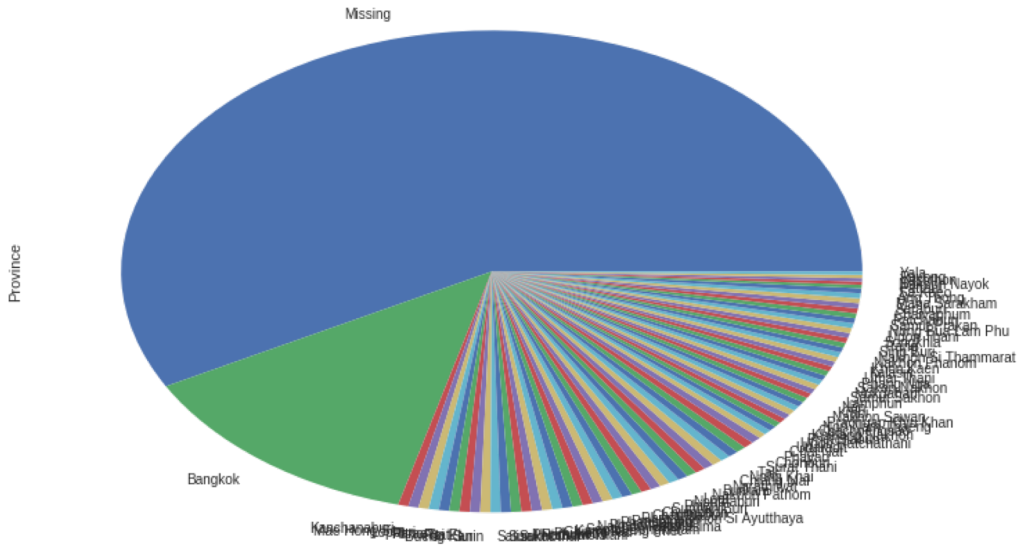
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f73cccc7fd0>
```



After

3) Drop records with missing >50%

```
df['Province'].fillna('Missing').value_counts().plot(kind='pie')
print( df['Province'].isnull().sum() )
print( float(df['Province'].isnull().sum()) / df.shape[0] )
```



	Gender	Payment Method	Churn	Country	Province	BirthYear
427	male	credit card	loyal	TH	NaN	1996.0
428	male	credit card	loyal	TH	NaN	1978.0
429	male	credit card	loyal	TH	NaN	1938.0
431	female	credit card	loyal	TH	NaN	1967.0
432	female	cash	churn	TH	NaN	1975.0

	Gender	Payment Method	Churn	Country	BirthYear
0	male	credit card	loyal	TH	1899.0
1	female	credit card	churn	TH	2000.0
2	male	credit card	loyal	TH	1980.0
3	female	credit card	churn	TH	1899.0
4	female	credit card	churn	TH	1985.0

4) Drop single value inputs

```
df = df.drop(['Country'], axis=1)
```

	Gender	Payment Method	Churn	Country	BirthYear
0	male	credit card	loyal	TH	1899.0
1	female	credit card	churn	TH	2000.0
2	male	credit card	loyal	TH	1980.0
3	female	credit card	churn	TH	1899.0
4	female	credit card	churn	TH	1985.0

Before

	Gender	Payment Method	Churn	BirthYear
0	male	credit card	loyal	1899.0
1	female	credit card	churn	2000.0
2	male	credit card	loyal	1980.0
3	female	credit card	churn	1899.0
4	female	credit card	churn	1985.0

After

5) Impute missing values: Numerical variable (mean)

```
df['BirthYear'] = df['BirthYear'].fillna(mean)
```

```
df[df['BirthYear'].isnull()].head(5)
```

	Gender	Payment Method	Churn	BirthYear
113	female	credit card	loyal	NaN
114	NaN	cash	loyal	NaN
115	male	credit card	loyal	NaN
116	female	credit card	loyal	NaN
118	male	cash	loyal	NaN

Before

```
#No missing records
df[df['BirthYear'].isnull()].head(5)
```

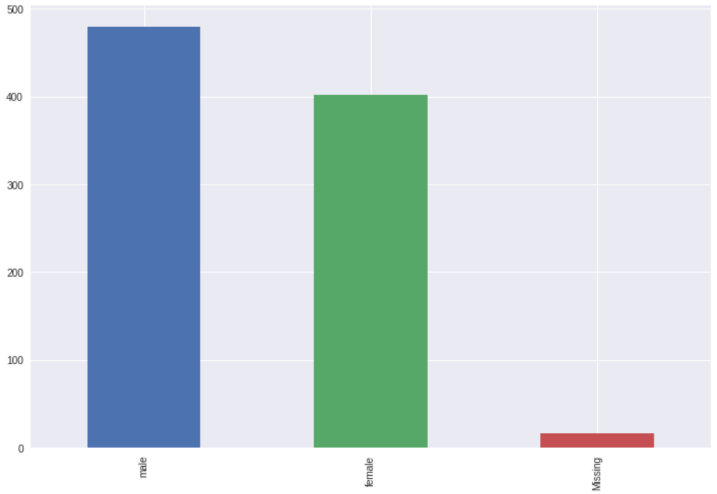
	Gender	Payment Method	Churn	BirthYear

After

5) Impute missing values: Categorical variable (mode)

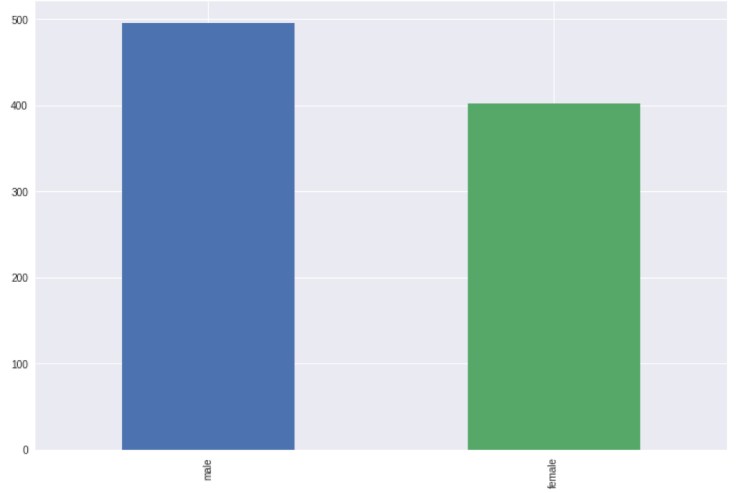
```
df['Gender'] = df['Gender'].fillna(mode)
```

```
df['Gender'].fillna('Missing').value_counts().plot(kind='bar')
<matplotlib.axes._subplots.AxesSubplot at 0x7f229071f710>
```



Before

```
df['Gender'].fillna('Missing').value_counts().plot(kind='bar')
<matplotlib.axes._subplots.AxesSubplot at 0x7f22904a7c50>
```



After

6) Convert non-numeric to numeric variable: Nominal variable using "apply"

```
conv_dict={'male':0,'female':1}
df['Gender']=df['Gender'].apply(conv_dict.get)
```

	Gender	Payment Method	Churn	BirthYear
0	male	credit card	loyal	1899.0
1	female	credit card	churn	2000.0
2	male	credit card	loyal	1980.0
3	female	credit card	churn	1899.0
4	female	credit card	churn	1985.0

Before

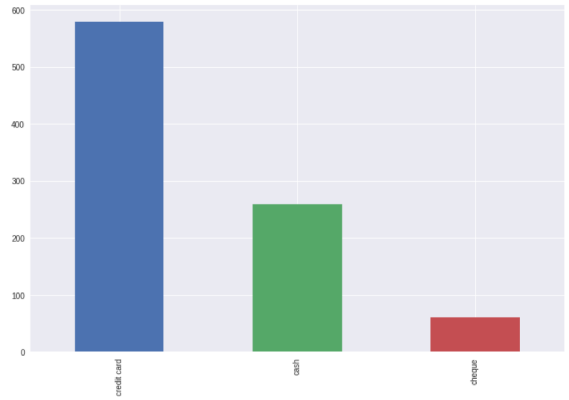
	Gender	Payment Method	Churn	BirthYear
0	0	credit card	loyal	1899.0
1	1	credit card	churn	2000.0
2	0	credit card	loyal	1980.0
3	1	credit card	churn	1899.0
4	1	credit card	churn	1985.0

After

6) Convert non-numeric to numeric variable: Nominal variable using "get_dummies"

```
payMtd_df = pd.get_dummies(df['Payment Method'], prefix='PayMethd')
payMtd_df.head(5)
```

	Gender	Payment Method	Churn	BirthYear
0	0	credit card	loyal	1899.0
1	1	credit card	churn	2000.0
2	0	credit card	loyal	1980.0
3	1	credit card	churn	1899.0
4	1	credit card	churn	1985.0




```
Payment Method
cash            259
cheque           60
credit card     579
dtype: int64
```

Before

After

	Gender	Churn	BirthYear	PayMethd_cash	PayMethd_cheque	PayMethd_credit card
0	0	loyal	1899.0	0	0	1
1	1	churn	2000.0	0	0	1
2	0	loyal	1980.0	0	0	1
3	1	churn	1899.0	0	0	1
4	1	churn	1985.0	0	0	1

7) Create calculated variables

```
now = datetime.datetime.today().year
now
```

```
2018
```

```
df['age'] = now - df['BirthYear']
```

Before

	Gender	Churn	BirthYear	PayMethd_cash	PayMethd_cheque	PayMethd_credit card
0	0	loyal	1899.0	0	0	1
1	1	churn	2000.0	0	0	1
2	0	loyal	1980.0	0	0	1
3	1	churn	1899.0	0	0	1
4	1	churn	1985.0	0	0	1

After

	Gender	Churn	PayMethd_cash	PayMethd_cheque	PayMethd_credit card	age
0	0	loyal	0	0	1	119.0
1	1	churn	0	0	1	18.0
2	0	loyal	0	0	1	38.0
3	1	churn	0	0	1	119.0
4	1	churn	0	0	1	33.0

8) Remove outlier

```
df['z_age'] = stats.zscore(df['age'])
```

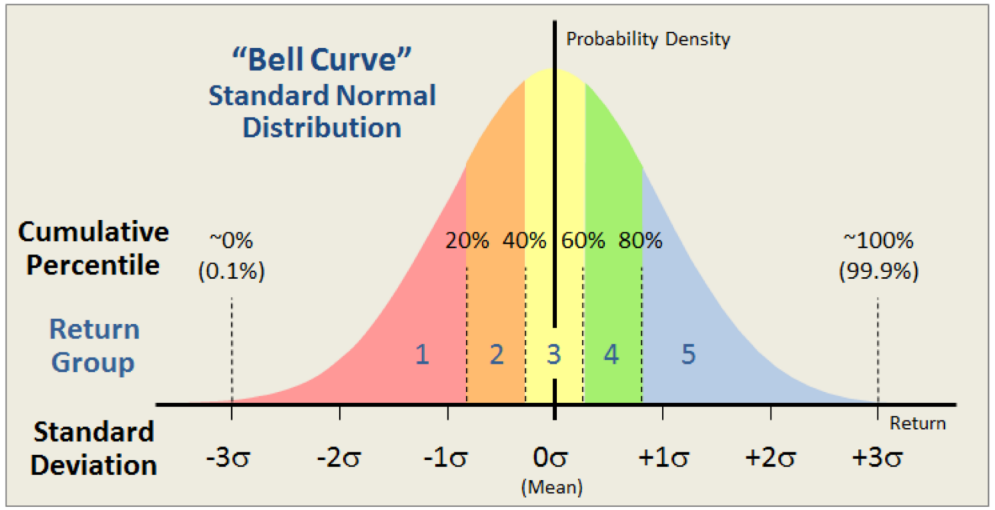
```
df['z_age'].min()
```

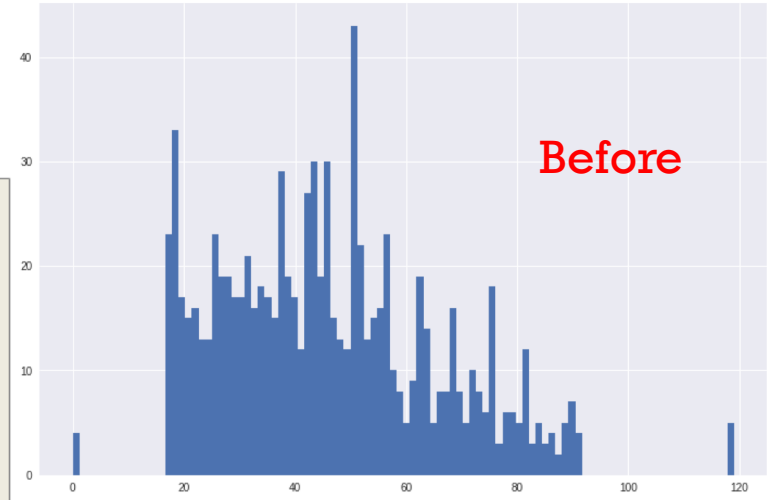
```
-2.3425085010581945
```

```
df['z_age'].max()
```

```
3.718779749022456
```

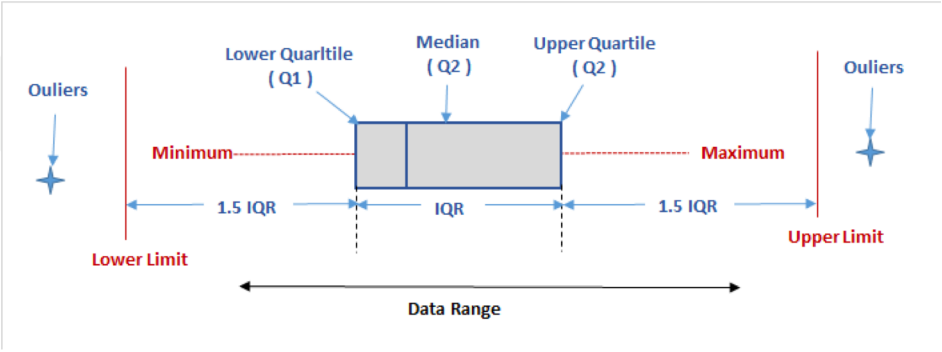
```
df = df[df['z_age'].abs() < 3]
```

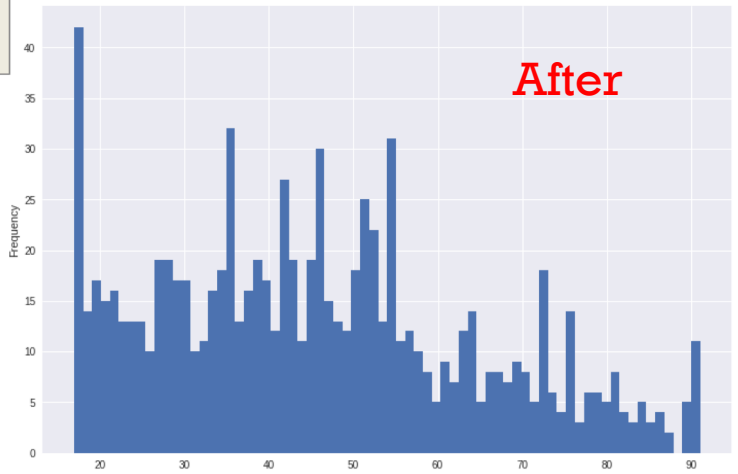





```
df['age'].plot.hist(bins=100, alpha=1)
```

```
df = df[df['age']> 0]
```





9) Save data

```
df.to_csv('customer_churn_cleaned.csv', sep=',', encoding='utf-8')
```

```
!ls
```

```
customer_churn_cleaned.csv  customer-churn-data_for-data-prep.csv  datalab
```

```
!head -5 customer_churn_cleaned.csv
```

```
,Gender,Churn,PayMethd_cash,PayMethd_cheque,PayMethd_credit card,age
1,1,churn,0,0,1,18.0
2,0,loyal,0,0,1,38.0
4,1,churn,0,0,1,33.0
5,0,churn,0,0,1,36.0
```

Other data preparation processes

- Impute missing values
- Outlier detections
- Feature transformation
  - Skewness
- Split train/test
  - Simple random sampling
  - Stratification
- Feature clustering
- Feature selection

Feature engineering

- Feature engineering
  - Calculated variables
  - Behavior from transactional data (RFM/RFA)

Recency	Frequency	Monetary Value
		
The time when they last placed an order	How many orders they have placed in the given period	How much money have they spent since their first purchase (CLV/LTV)

+

Any Questions?