Introduction to Data Science
CS61
June 12 - July 12, 2018

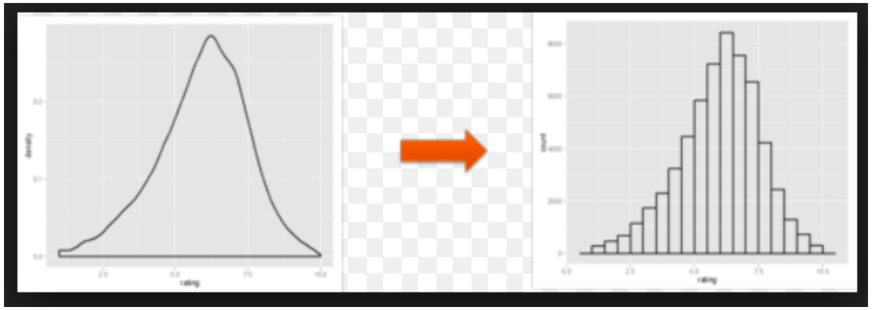
Dr. Ash Pahwa

Lesson 3: Data Exploration-2

Lesson 3.1: Discretization

Outline

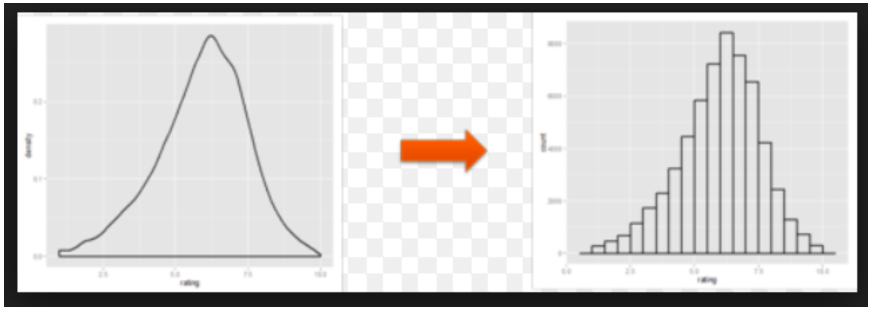
- Discretization
- Discretization in R
- Discretization in Python



Discretization

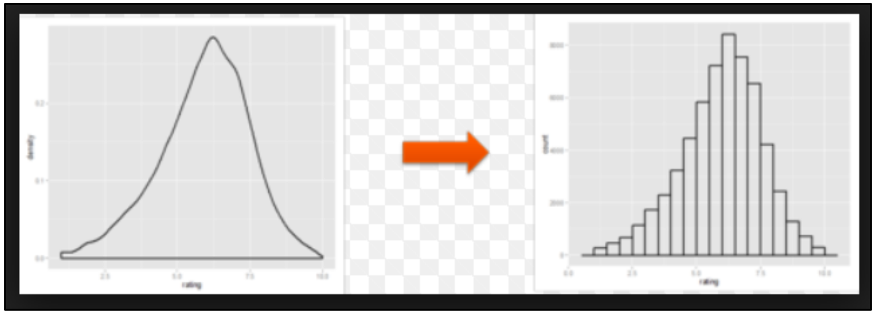
Converting Numeric Data into Categorical Data

Why Discretization?



- Data Mining algorithms
  - Classification Modeling Methods:
    - Deals with categorical data (nominal)
  - Example: Decision Trees + Naïve Bayes
- We need to convert numeric attributes into small number of distinct ranges or categorical values

Why Discretization?



- When the observed data is not very precise
  - We discretize the data

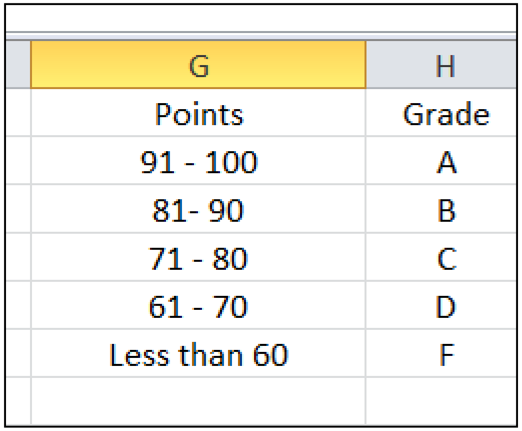
- Annual Income
  - Less than $20,000: Poor
  - Between $20,000 - $40,000: Lower middle class
  - Between $40,000 - $80,000: Upper middle class
  - Between $80,000 - $200,000: Rich
  - Between $200,000 - $1,000,000: Very Rich
  - Above $1,000,000: Super Rich

Example of Discretization

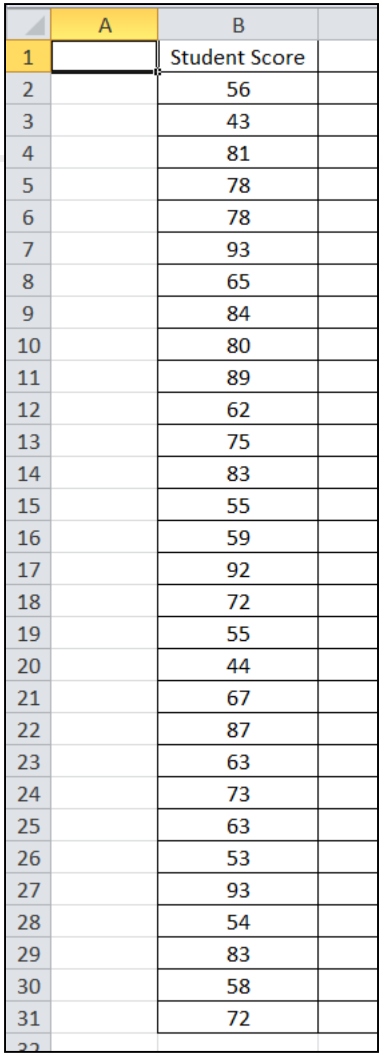
Numeric to Categorical

Student’s Earned Points Converted to Grades

- Total 30 students
- Score
  - From 1 to 100



	G	H
	Points	Grade
	91 - 100	A
	81- 90	B
	71 - 80	C
	61 - 70	D
	Less than 60	F



	A	B
1		Student Score
2		56
3		43
4		81
5		78
6		78
7		93
8		65
9		84
10		80
11		89
12		62
13		75
14		83
15		55
16		59
17		92
18		72
19		55
20		44
21		67
22		87
23		63
24		73
25		63
26		53
27		93
28		54
29		83
30		58
31		72

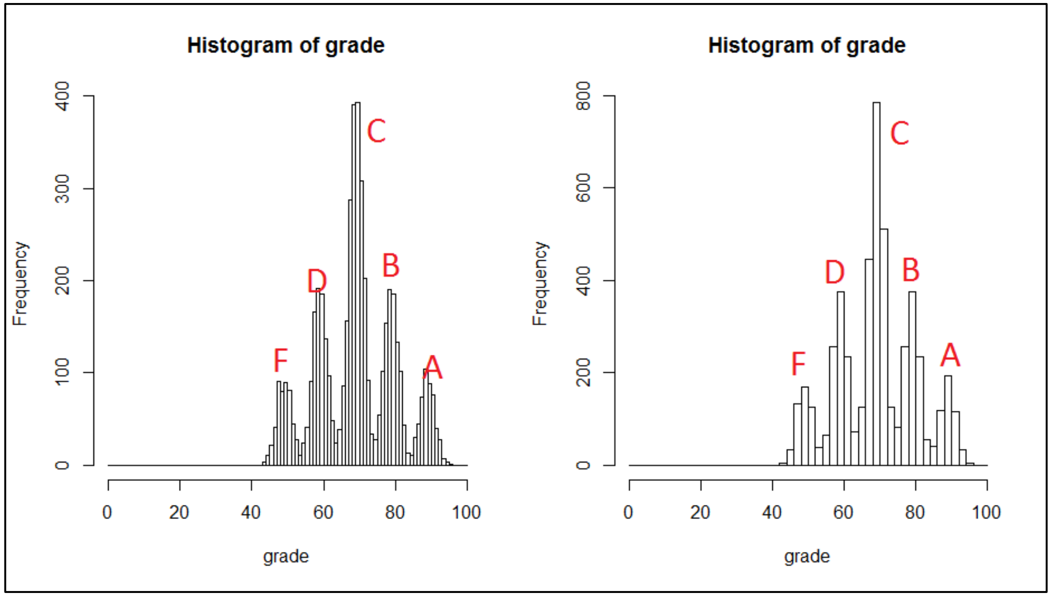
Discretization

- How to determine the boundaries between classes?
  - Natural boundaries
  - Equi-width ranges
  - Equi-log ranges
  - Equi-depth ranges

Natural Boundaries

- Students count = 5,000
- Histogram with different bin sizes
  - Bin size = 1, 2

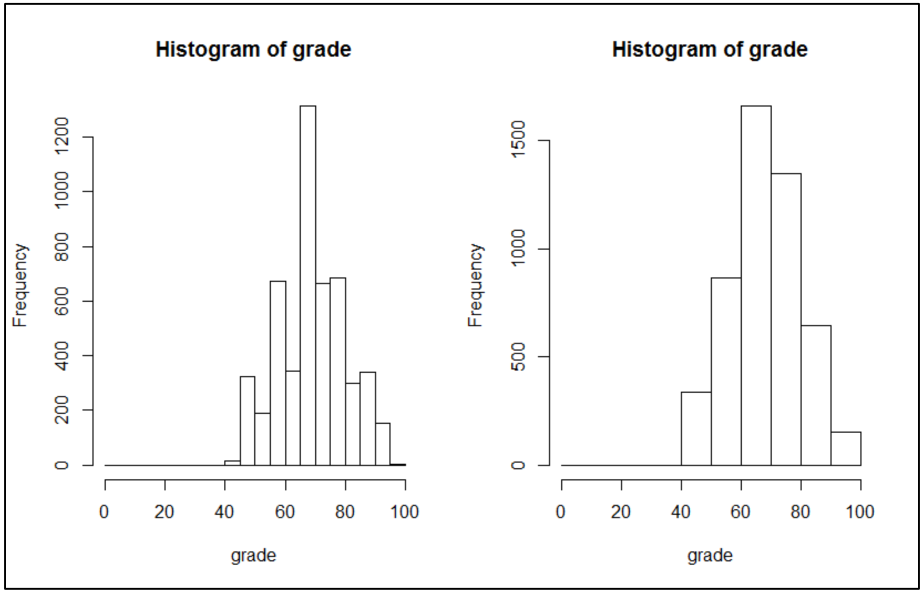
```
par(mfrow=c(2,2))
hist(grade,seq(0,100,1))
hist(grade,seq(0,100,2))
```



Equi-width Ranges

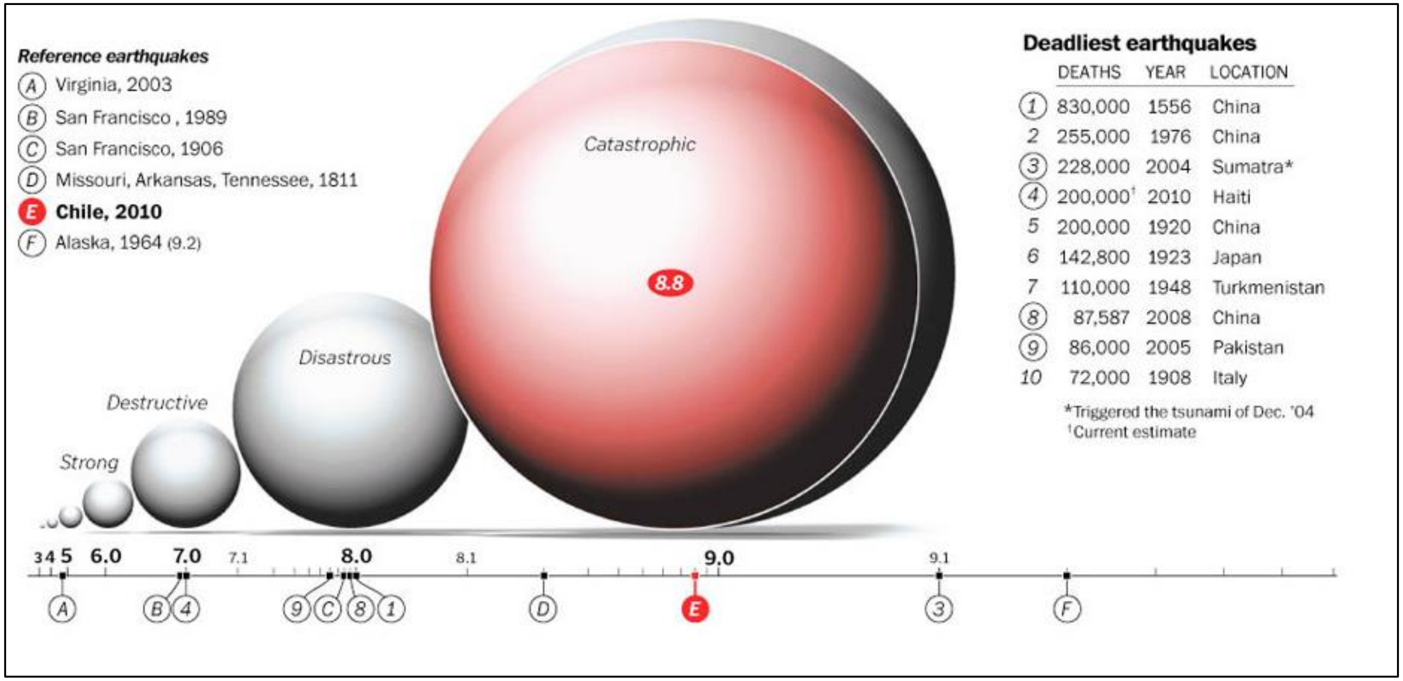
- Range [a,b] is chosen
  - (b-a) = constant for all ranges
  - Will not work if data is non-uniformly distributed
- Range Fixed = 10
  - Points 90-100 = A
  - Points 80-89 = B
  - Points 70–79 = C
  - Points 60-69 = D
  - Points < 60 = F

Histograms with bin size = 5 and 10



Equi-log Ranges

- Range [a,b] is chosen
  - (log(b)-log(a)) = constant for all ranges
  - Works for exponentially distributed data



Deadliest earthquakes

	DEATHS	YEAR	LOCATION
1	830,000	1556	China
2	255,000	1976	China
3	228,000	2004	Sumatra*
4	200,000[†]	2010	Haiti
5	200,000	1920	China
6	142,800	1923	Japan
7	110,000	1948	Turkmenistan
8	87,587	2008	China
9	86,000	2005	Pakistan
10	72,000	1908	Italy

*Triggered the tsunami of Dec. '04

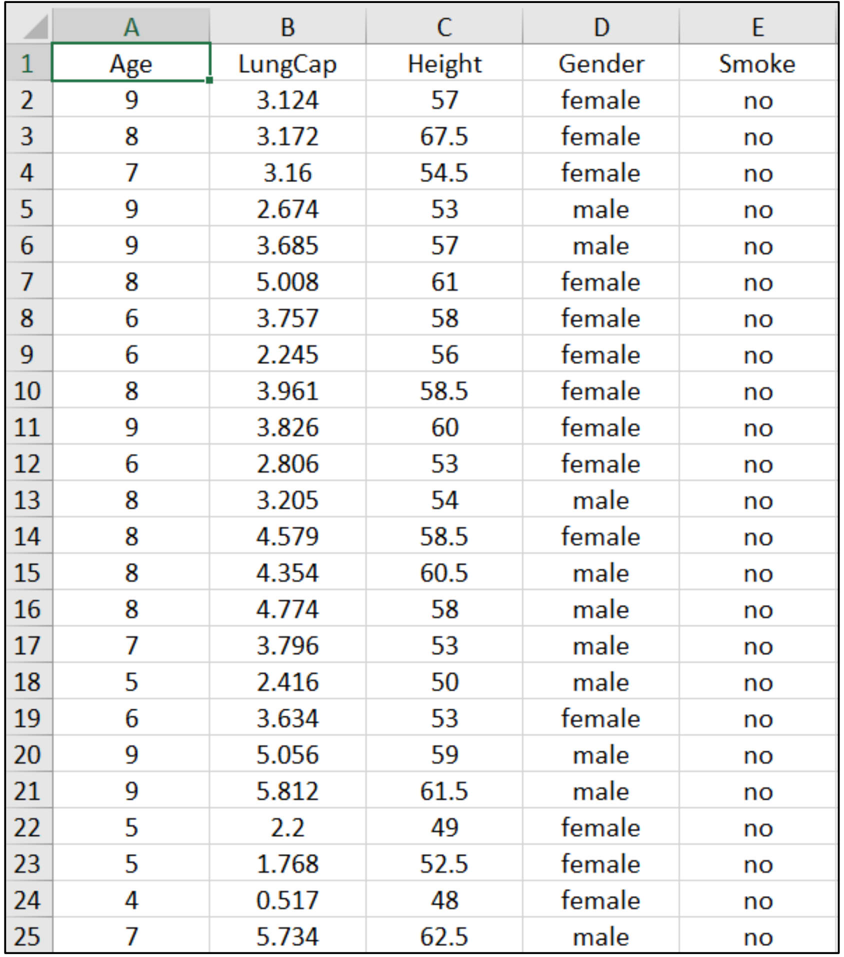
[†] Current estimate

Equi-depth Ranges

- Range [a,b] is chosen
  - Each range has an equal number of records
  - First sort the data
    - Select the boundaries from the sorted data such that each range contains equal number of observations

Raw Data
Lung Capacity Data

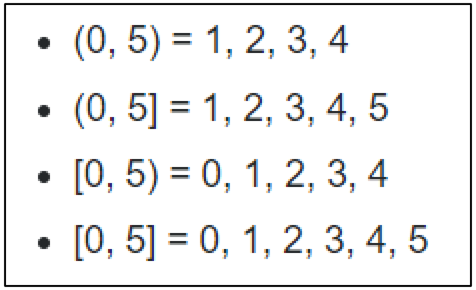
- Response Variable
  - Lung Capacity: Numerical
- Predictor Variables
  - Age: Numerical
  - Height: Numerical
  - Gender: Categorical
  - Smoke: Categorical



	A	B	C	D	E
1	Age	LungCap	Height	Gender	Smoke
2	9	3.124	57	female	no
3	8	3.172	67.5	female	no
4	7	3.16	54.5	female	no
5	9	2.674	53	male	no
6	9	3.685	57	male	no
7	8	5.008	61	female	no
8	6	3.757	58	female	no
9	6	2.245	56	female	no
10	8	3.961	58.5	female	no
11	9	3.826	60	female	no
12	6	2.806	53	female	no
13	8	3.205	54	male	no
14	8	4.579	58.5	female	no
15	8	4.354	60.5	male	no
16	8	4.774	58	male	no
17	7	3.796	53	male	no
18	5	2.416	50	male	no
19	6	3.634	53	female	no
20	9	5.056	59	male	no
21	9	5.812	61.5	male	no
22	5	2.2	49	female	no
23	5	1.768	52.5	female	no
24	4	0.517	48	female	no
25	7	5.734	62.5	male	no

Square Brackets and Parenthesis

- A square bracket means that end of the range is inclusive –
  - It includes the element listed.
- A parenthesis means that end is exclusive and doesn't contain the listed element.
- So for [first1, last1), the range starts with first1 (and includes it), but ends just before last1.



- (0, 5) = 1, 2, 3, 4
- (0, 5] = 1, 2, 3, 4, 5
- [0, 5) = 0, 1, 2, 3, 4
- [0, 5] = 0, 1, 2, 3, 4, 5

Discretization in R

Discretize Height into 4 Equi-width categories

Height inches	Category
(50 – 54.4]	A
(54.4 – 58.8]	B
(58.8 – 63.1]	C
(63.1 – 67.5]	D

```
> LungCapData
   Age LungCap Height Gender Smoke
1     9   3.124   57.0 female    no
2     8   3.172   67.5 female    no
3     7   3.160   54.5 female    no
4     9   2.674   53.0   male    no
5     9   3.685   57.0   male    no
6     8   5.008   61.0 female    no
7     6   3.757   58.0 female    no
8     6   2.245   56.0 female    no
9     8   3.961   58.5 female    no
10    9   3.826   60.0 female    no
11    6   2.806   53.0 female    no
12    8   3.205   54.0   male    no
13    8   4.579   58.5 female    no
14    8   4.354   60.5   male    no
15    8   4.774   58.0   male    no
16    7   3.796   53.0   male    no
17    5   2.416   50.0   male    no
18    6   3.634   53.0 female    no
19    9   5.056   59.0   male    no
20    9   5.812   61.5   male    no
```

```
> NumCategories = 4
> min(LungCapData$Height)
[1] 50
> max(LungCapData$Height)
[1] 67.5
> (range = max(LungCapData$Height) -
min(LungCapData$Height))
[1] 17.5
> (binWidth = range/NumCategories)
[1] 4.375
> (bin1Upper = min(LungCapData$Height) + binWidth)
[1] 54.375
> (bin2Upper = bin1Upper + binWidth)
[1] 58.75
> (bin3Upper = bin2Upper + binWidth)
[1] 63.125
> (bin4Upper = bin3Upper + binWidth)
[1] 67.5
```

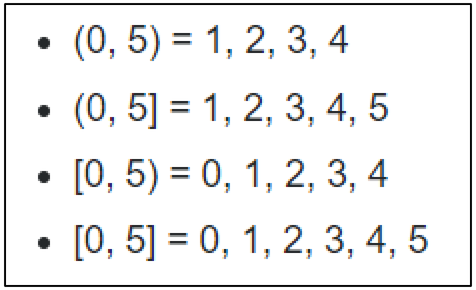
Discretize Height into 4 Equi-width categories

```
> LungCapData
   Age LungCap Height Gender Smoke
1     9   3.124   57.0 female    no
2     8   3.172   67.5 female    no
3     7   3.160   54.5 female    no
4     9   2.674   53.0   male    no
5     9   3.685   57.0   male    no
6     8   5.008   61.0 female    no
7     6   3.757   58.0 female    no
8     6   2.245   56.0 female    no
9     8   3.961   58.5 female    no
10    9   3.826   60.0 female    no
11    6   2.806   53.0 female    no
12    8   3.205   54.0   male    no
13    8   4.579   58.5 female    no
14    8   4.354   60.5   male    no
15    8   4.774   58.0   male    no
16    7   3.796   53.0   male    no
17    5   2.416   50.0   male    no
18    6   3.634   53.0 female    no
19    9   5.056   59.0   male    no
20    9   5.812   61.5   male    no
```

```
> class(LungCapData$Height)
[1] "numeric"
> hist(LungCapData$Height)
>
> NumCategories = 4
> (c1 = cut(LungCapData$Height,breaks=NumCategories))
 [1] (54.4,58.8] (63.1,67.5] (54.4,58.8] (50,54.4]
 [5] (54.4,58.8] (58.8,63.1] (54.4,58.8] (54.4,58.8]
 [9] (54.4,58.8] (58.8,63.1] (50,54.4]   (50,54.4]
[13] (54.4,58.8] (58.8,63.1] (54.4,58.8] (50,54.4]
[17] (50,54.4]   (50,54.4]   (58.8,63.1] (58.8,63.1]
4 Levels: (50,54.4] (54.4,58.8] ... (63.1,67.5]
> (count1 = as.vector(table(c1)))
[1] 6 8 5 1
>
> class(c1)
[1] "factor"
> levels(c1)
[1] "(50,54.4]"   "(54.4,58.8]" "(58.8,63.1]"
[4] "(63.1,67.5]"
>
> LungCapData$Height[1:10]
 [1] 57.0 67.5 54.5 53.0 57.0 61.0 58.0 56.0 58.5 60.0
> c1[1:10]
 [1] (54.4,58.8] (63.1,67.5] (54.4,58.8] (50,54.4]
 [5] (54.4,58.8] (58.8,63.1] (54.4,58.8] (54.4,58.8]
 [9] (54.4,58.8] (58.8,63.1]
4 Levels: (50,54.4] (54.4,58.8] ... (63.1,67.5]
>
```

Discretize Height into 6 Categories : Width Size is Different

Height inches	Category
(0 – 50]	A
(50 – 55]	B
(55 – 60]	C
(60 – 65]	D
(65 – 70]	E
> 70	F



- (0, 5) = 1, 2, 3, 4
- (0, 5] = 1, 2, 3, 4, 5
- [0, 5) = 0, 1, 2, 3, 4
- [0, 5] = 0, 1, 2, 3, 4, 5

Discretize Height into 6 Categories : Width Size is Different

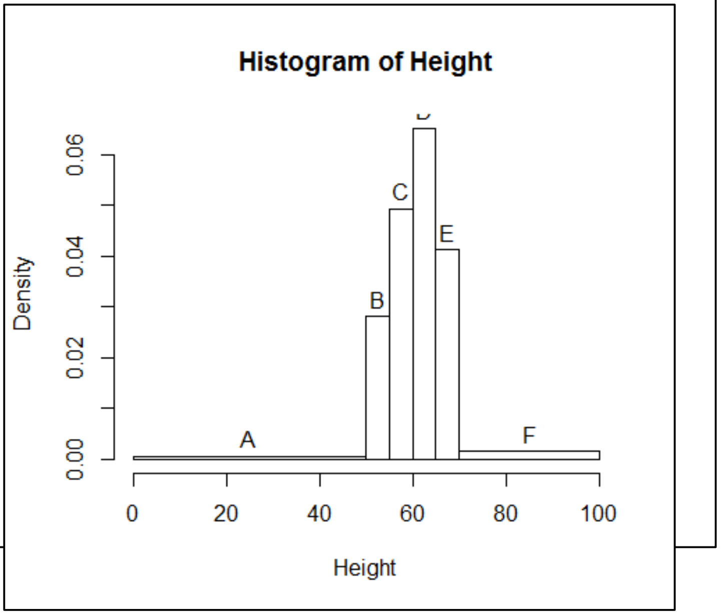
Height inches	Category
(0 – 50]	A
(50 – 55]	B
(55 – 60]	C
(60 – 65]	D
(65 – 70]	E
> 70	F

```
> hist(Height,breaks=c(0,50,55,60,65,70,100),labels=c("A","B","C","D","E","F"))

> catHeight = cut(Height, breaks=c(0,50,55,60,65,70,100),labels=c("A","B","C","D","E","F"))
> (count1 = as.vector(table(catHeight)))
[1]  22  92 161 213 135  31


> class(catHeight)
[1] "factor"
> levels(catHeight)
[1] "A" "B" "C" "D" "E" "F"


> Height[1:10]
 [1] 57.0 67.5 54.5 53.0 57.0 61.0 58.0 56.0 58.5 60.0
> catHeight[1:10]
 [1] C E B B C D C C C C
Levels: A B C D E F
>
```



Discretization in Python

Read Datafile

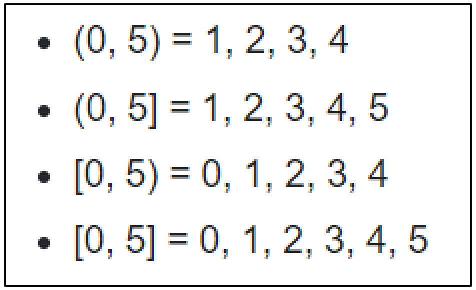
```
import pandas as pd

df = pd.read_csv('Lung Capacity.csv')

df[0:10]
Out[3]:
   Age  LungCap  Height  Gender Smoke
0    9    3.124    57.0  female    no
1    8    3.172    67.5  female    no
2    7    3.160    54.5  female    no
3    9    2.674    53.0    male    no
4    9    3.685    57.0    male    no
5    8    5.008    61.0  female    no
6    6    3.757    58.0  female    no
7    6    2.245    56.0  female    no
8    8    3.961    58.5  female    no
9    9    3.826    60.0  female    no
```

Discretize Height into 6 Categories : Width Size is Different

Height inches	Category
(0 – 50]	A
(50 – 55]	B
(55 – 60]	C
(60 – 65]	D
(65 – 70]	E
> 70	F



- (0, 5) = 1, 2, 3, 4
- (0, 5] = 1, 2, 3, 4, 5
- [0, 5) = 0, 1, 2, 3, 4
- [0, 5] = 0, 1, 2, 3, 4, 5

Discretize Height into 6 Categories : Width Size is Different

Height inches	Category
(0 – 50]	A
(50 – 55]	B
(55 – 60]	C
(60 – 65]	D
(65 – 70]	E
> 70	F

```
bins = [0, 50, 55, 60, 65, 70, 100]

group_names = ['A', 'B', 'C', 'D','E','F']

c1 = pd.cut(df['Height'], bins, labels=group_names)

print(c1.value_counts())
D     213
C     161
E     135
B      92
F      31
A      22
Name: Height, dtype: int64

dict(c1.value_counts())
Out[18]: {'A': 22, 'B': 92, 'C': 161, 'D': 213, 'E': 135, 'F': 31}
```

Discretized Data

Height inches	Category
(0 – 50]	A
(50 – 55]	B
(55 – 60]	C
(60 – 65]	D
(65 – 70]	E
> 70	F

```
print( df['Height'][0:8] )
0     57.0
1     67.5
2     54.5
3     53.0
4     57.0
5     61.0
6     58.0
7     56.0
Name: Height, dtype: float64

print( c1[0:8] )
0     C
1     E
2     B
3     B
4     C
5     D
6     C
7     C
Name: Height, dtype: category
Categories (6, object): [A < B < C < D < E < F]
```

```
df['grade'] = pd.cut(df['Height'], bins,
labels=group_names)

df[0:8]
Out[26]:
   Age  LungCap  Height  Gender Smoke grade
0    9    3.124    57.0  female    no     C
1    8    3.172    67.5  female    no     E
2    7    3.160    54.5  female    no     B
3    9    2.674    53.0    male    no     B
4    9    3.685    57.0    male    no     C
5    8    5.008    61.0  female    no     D
6    6    3.757    58.0  female    no     C
7    6    2.245    56.0  female    no     C
```

Summary

- Discretization
- Discretization in R
- Discretization in Python