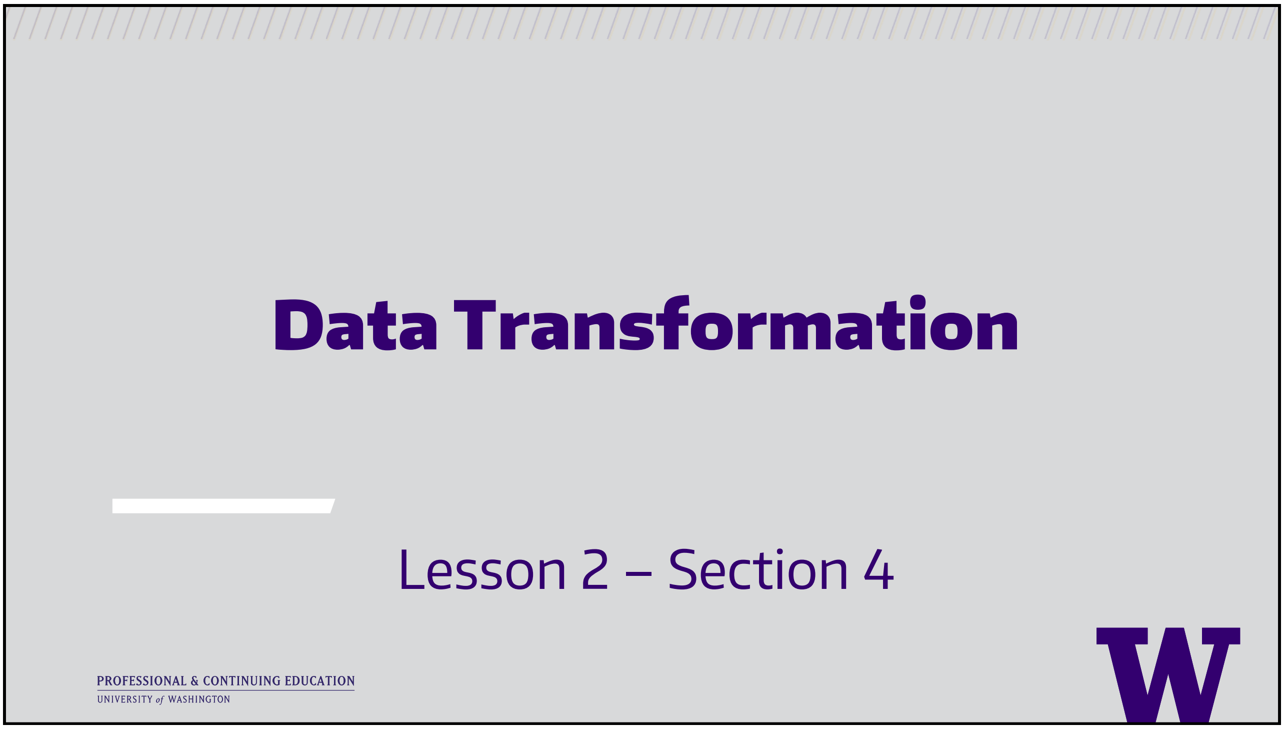
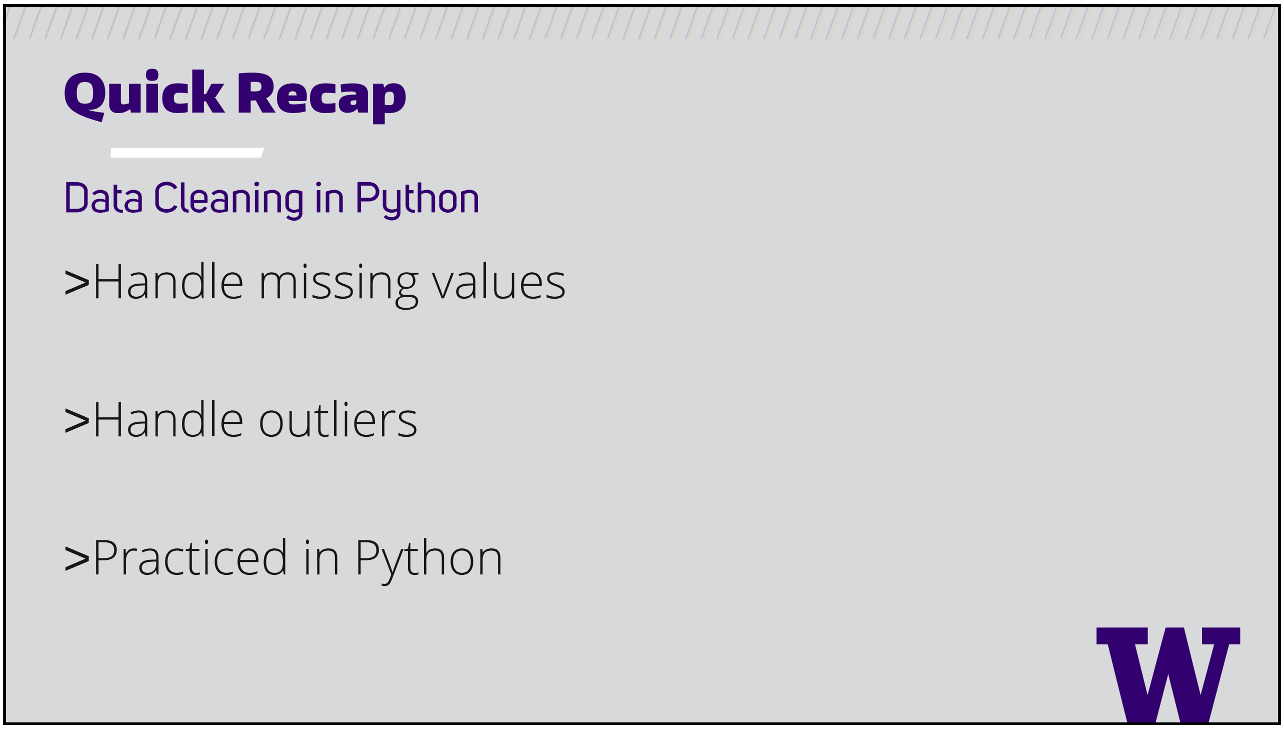


Data Transformation

Lesson 2 – Section 4

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON



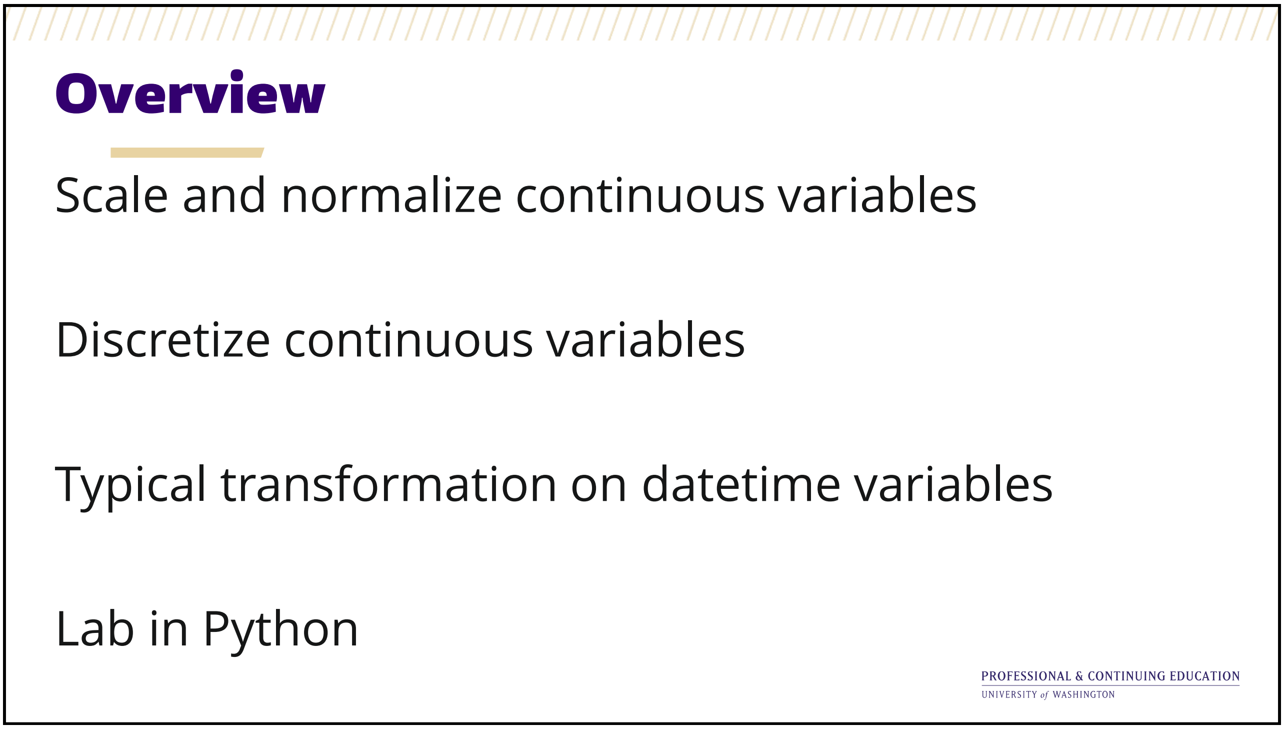
Quick Recap

Data Cleaning in Python

>Handle missing values

>Handle outliers

>Practiced in Python



Overview

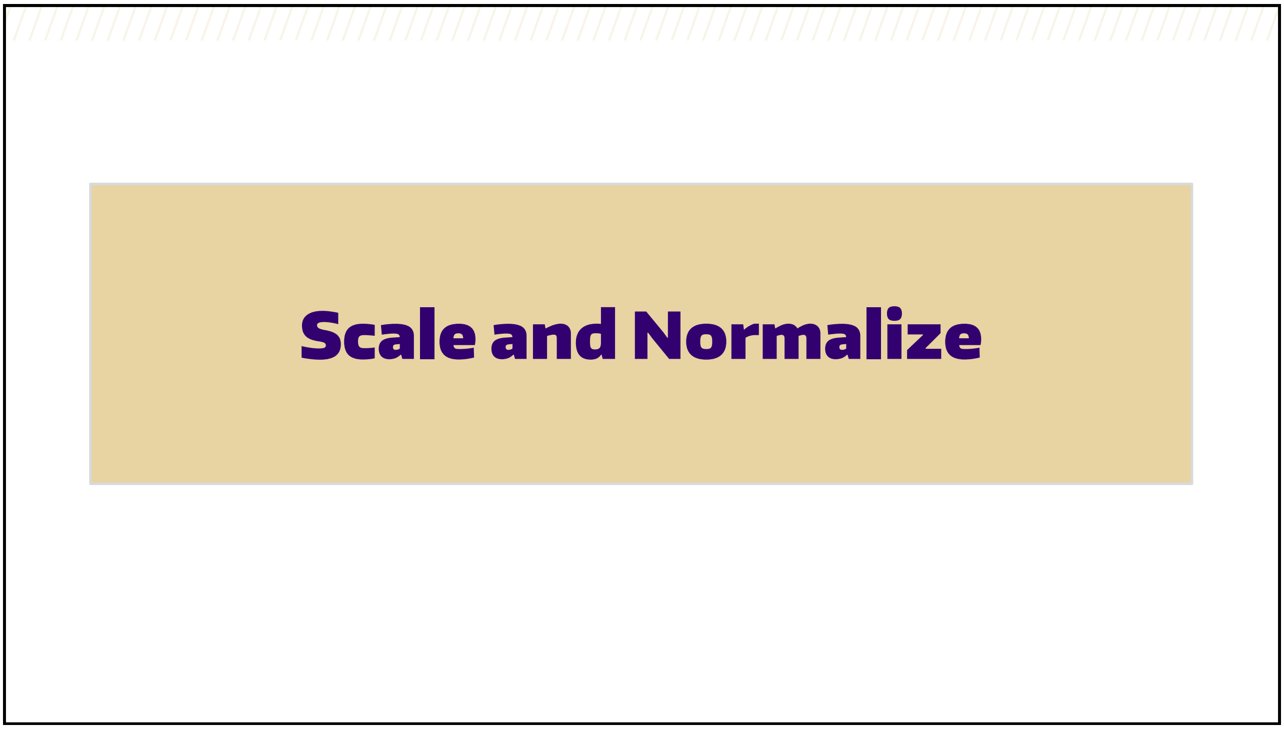
Scale and normalize continuous variables

Discretize continuous variables

Typical transformation on datetime variables

Lab in Python

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON



Scale and Normalize

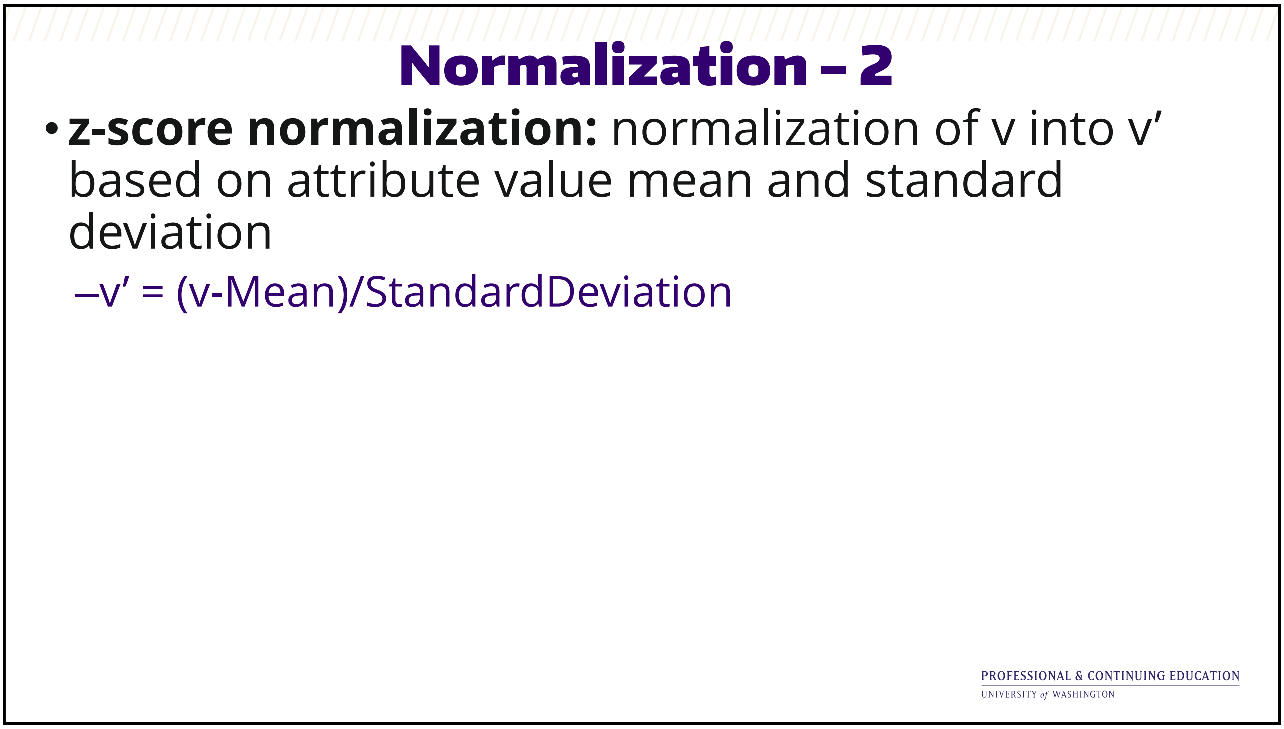
Scaling of Continuous Variables

- Many ML algorithms rely on measuring the distance between 2 samples
- There should be no difference if a length variable is measured in cm, inch, or km
- To remove the unit of measure (e.g. kg, mph, …) each variable dimension is normalized:
  - subtract mean
  - divide by standard deviation

Normalization – 1

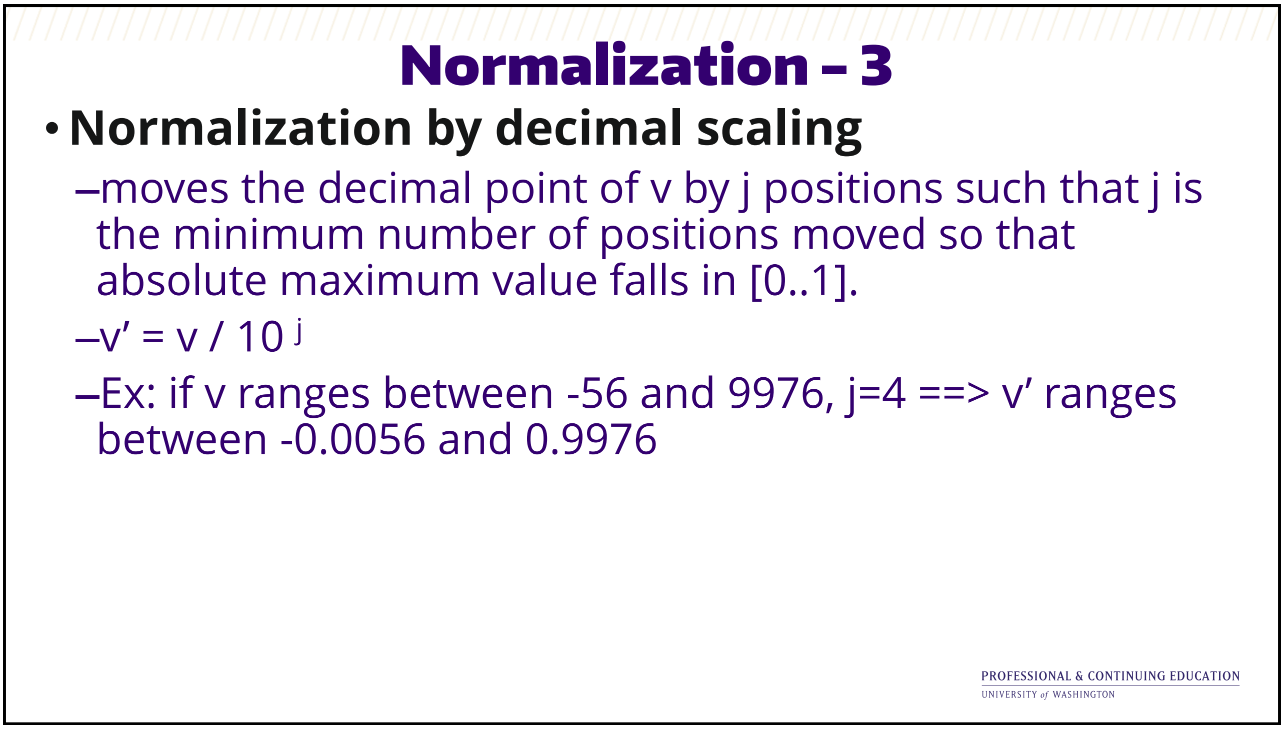
- **Min-max normalization:** linear transformation from v to v'
  - v' = (v – min)/((max - min)*(newmax - newmin)) + new min
  - Ex: transform $30000 between [10000..45000] into [0..1]

    ==> (30 – 10)/(35(1)) + 0 = 0.5714



Normalization – 2

- **z-score normalization:** normalization of v into v' based on attribute value mean and standard deviation
  - v' = (v-Mean)/StandardDeviation



Normalization – 3

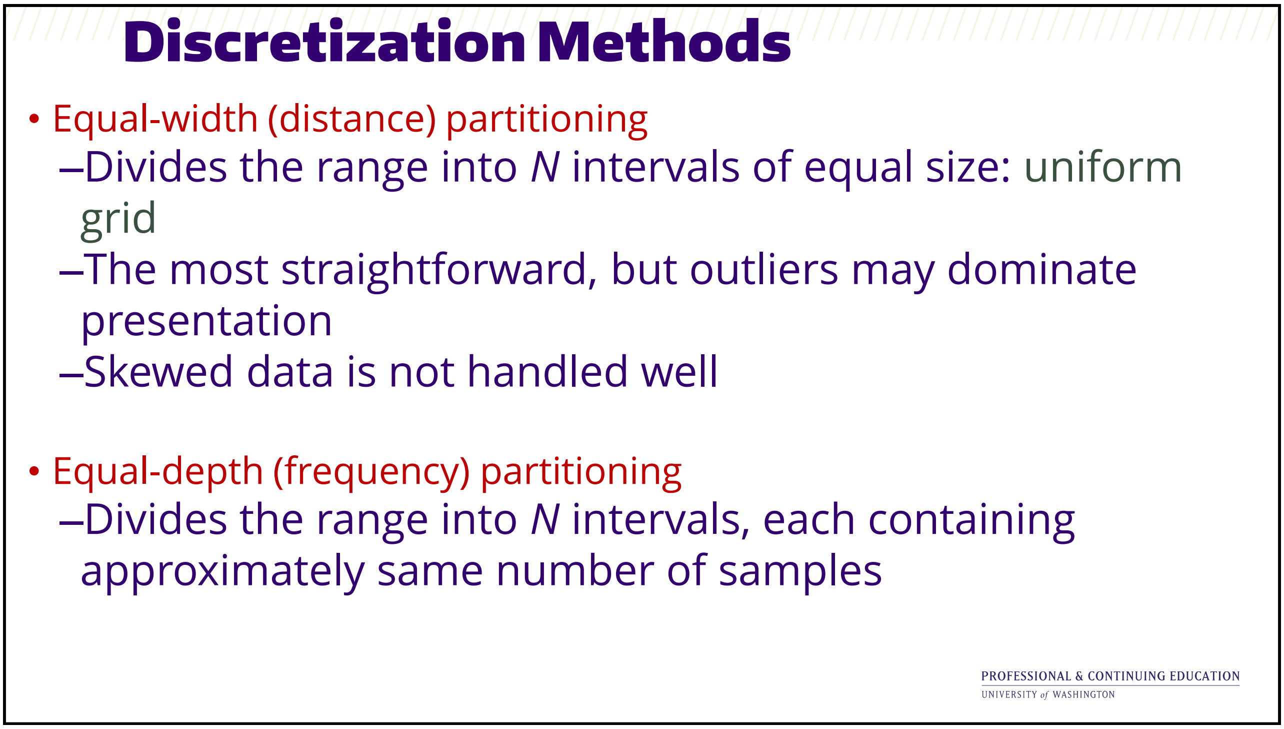
- **Normalization by decimal scaling**
  - moves the decimal point of v by j positions such that j is the minimum number of positions moved so that absolute maximum value falls in [0..1].
  - v' = v / 10 ^{j}
  - Ex: if v ranges between -56 and 9976, j=4 ==> v' ranges between -0.0056 and 0.9976

Discretize Continuous Variables

Discretization/Binning

Less features, more discrimination ability

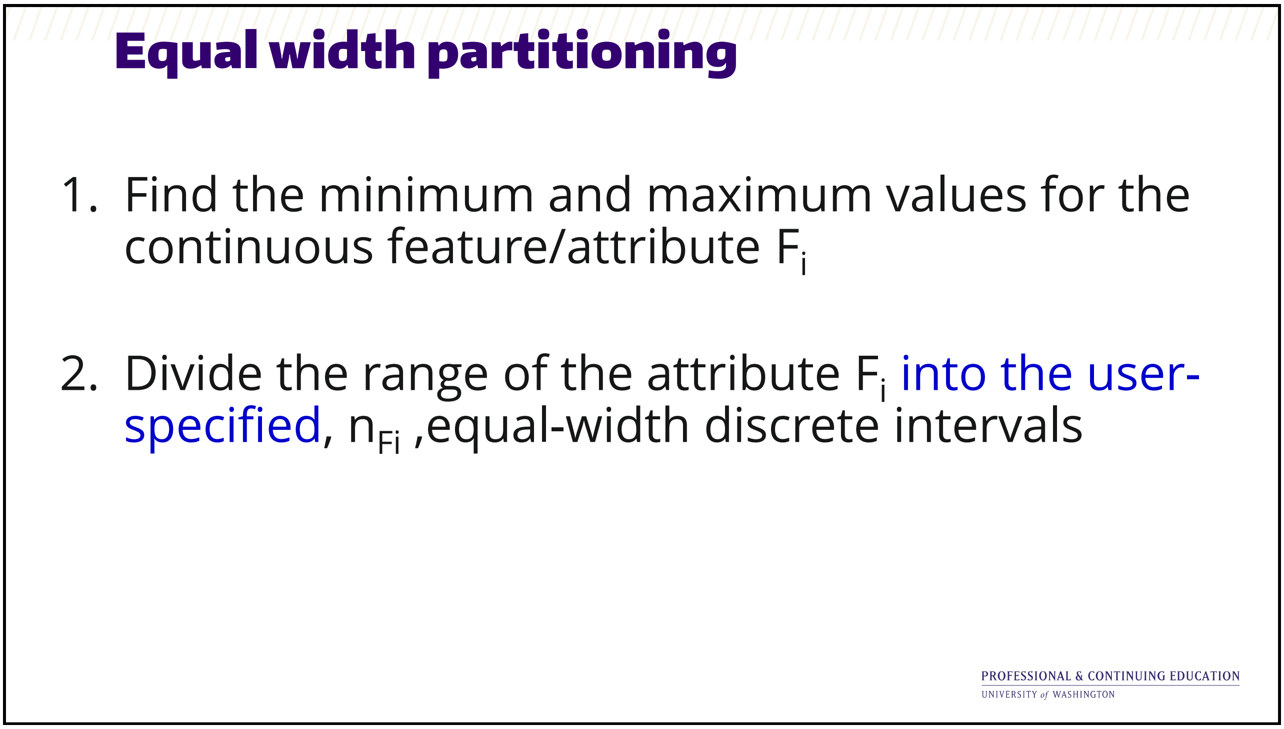
- Discretization is used to reduce the number of values for a given continuous attribute
  - –usually done by dividing the range of the attribute into intervals
  - –interval labels are then used to replace actual data values
- Discretization can also be used to generate concept hierarchies
  - –reduce the data by collecting and replacing low level concepts (e.g., numeric values for “age”) by higher level concepts (e.g., “young”, “middle aged”, “old”)



Discretization Methods

- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples

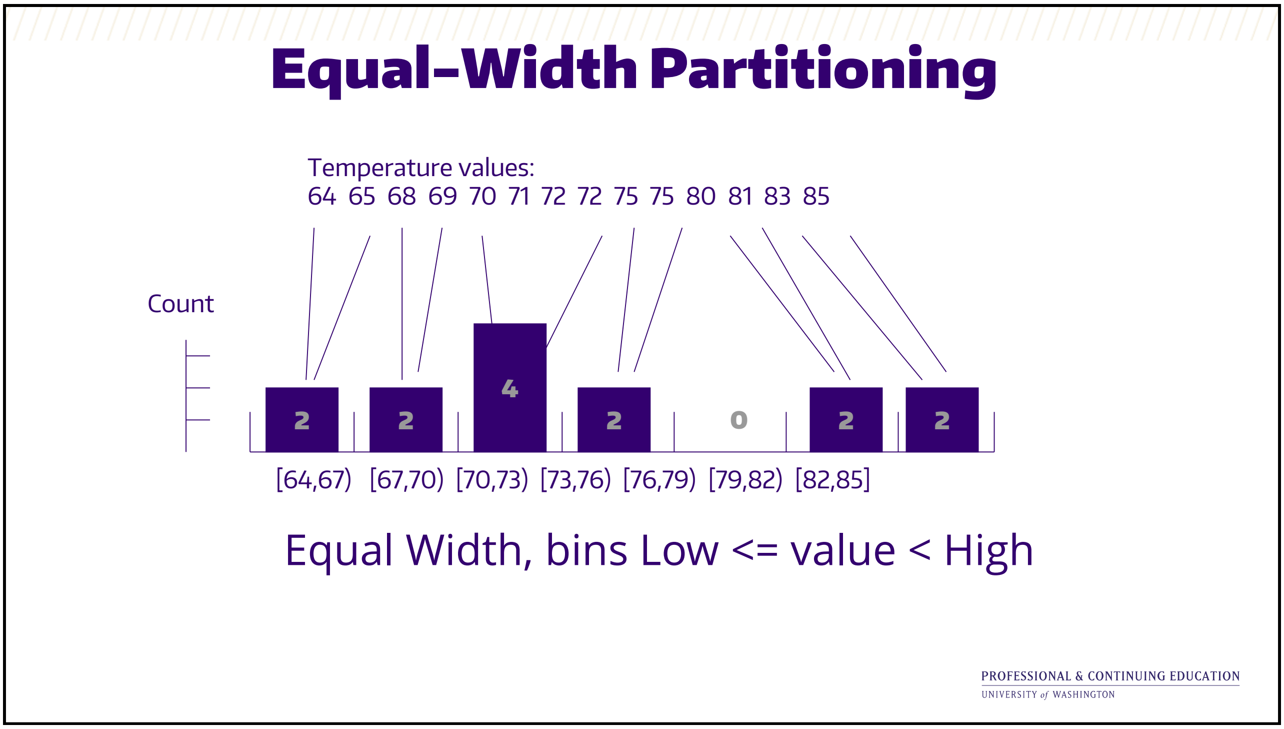
PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY *of* WASHINGTON



Equal width partitioning

1. Find the minimum and maximum values for the continuous feature/attribute F_i
2. Divide the range of the attribute F_i into the user-specified, n_{Fi} ,equal-width discrete intervals

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY *of* WASHINGTON

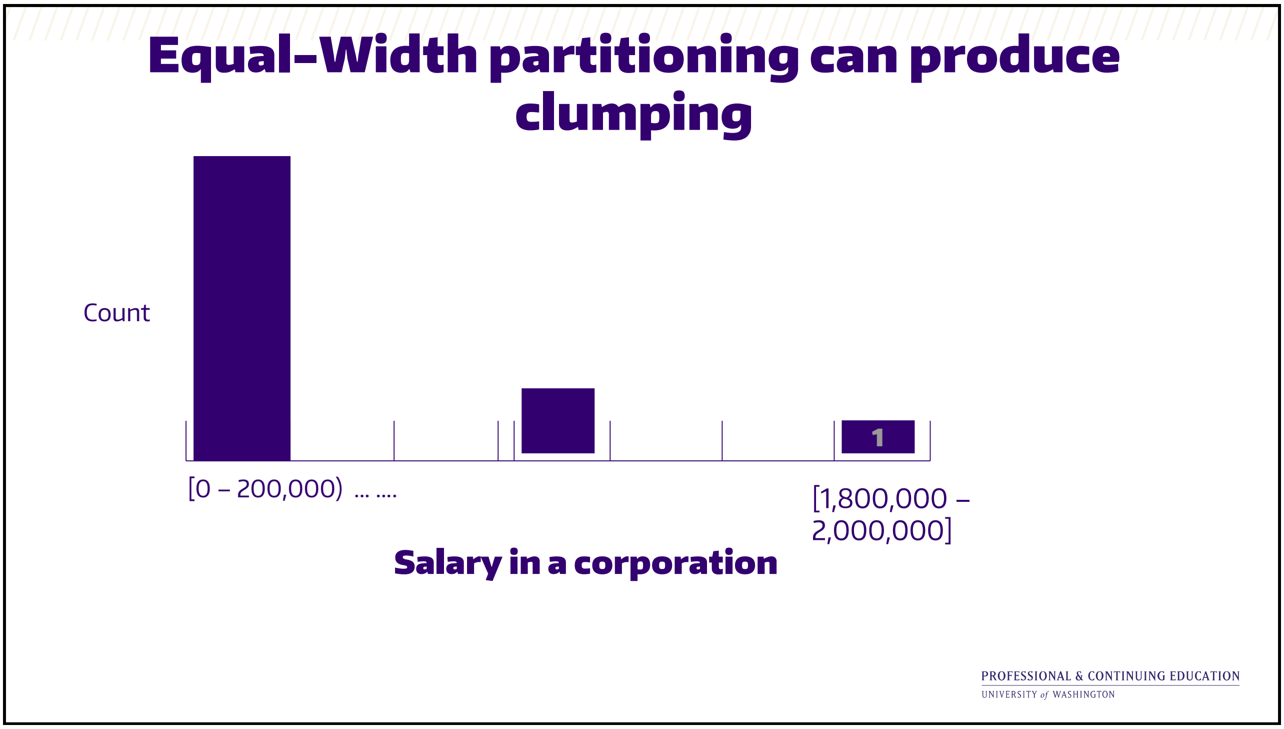


Equal-Width Partitioning

Temperature values:
64 65 68 69 70 71 72 72 75 75 80 81 83 85

Equal Width, bins Low <= value < High

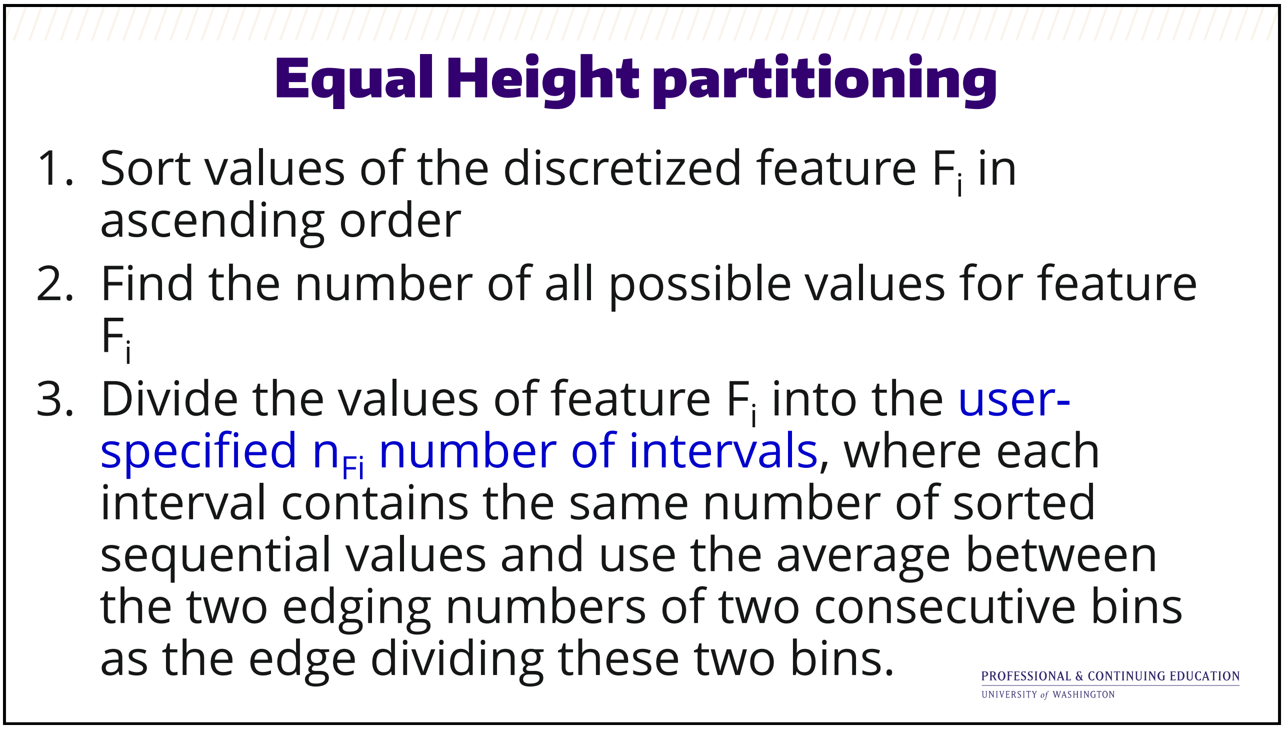
PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON



Equal-Width partitioning can produce clumping

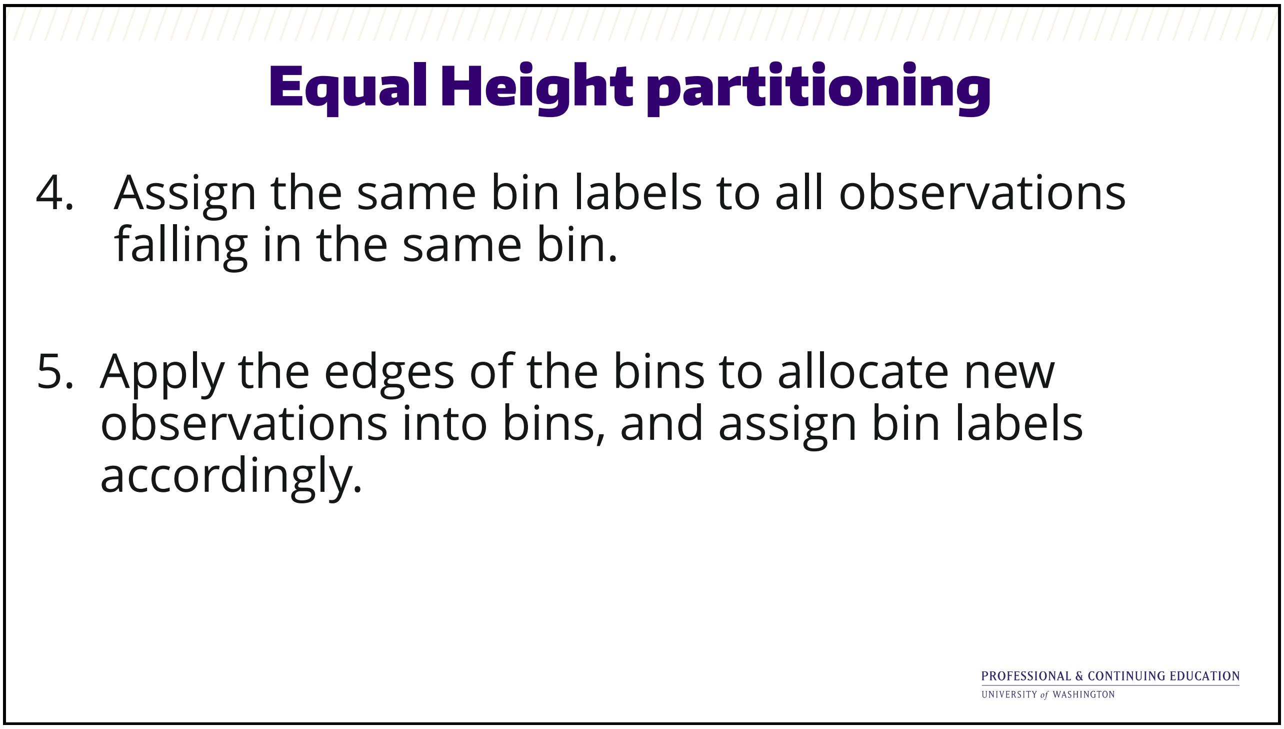
Salary in a corporation

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON



Equal Height partitioning

1. Sort values of the discretized feature F_i in ascending order
2. Find the number of all possible values for feature F_i
3. Divide the values of feature F_i into the user-specified n_{Fi} number of intervals, where each interval contains the same number of sorted sequential values and use the average between the two edging numbers of two consecutive bins as the edge dividing these two bins.



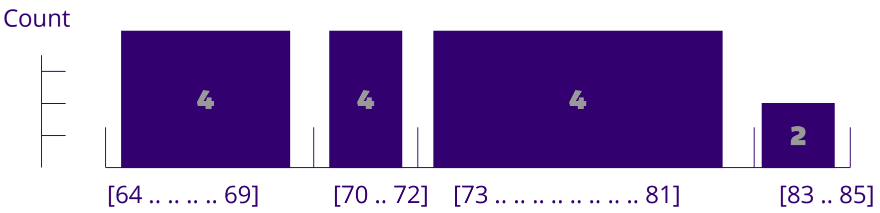
Equal Height partitioning

4. Assign the same bin labels to all observations falling in the same bin.

5. Apply the edges of the bins to allocate new observations into bins, and assign bin labels accordingly.

Equal-Height partitioning

Temperature values:
64 65 68 69 70 71 72 72 75 75 80 81 83 85



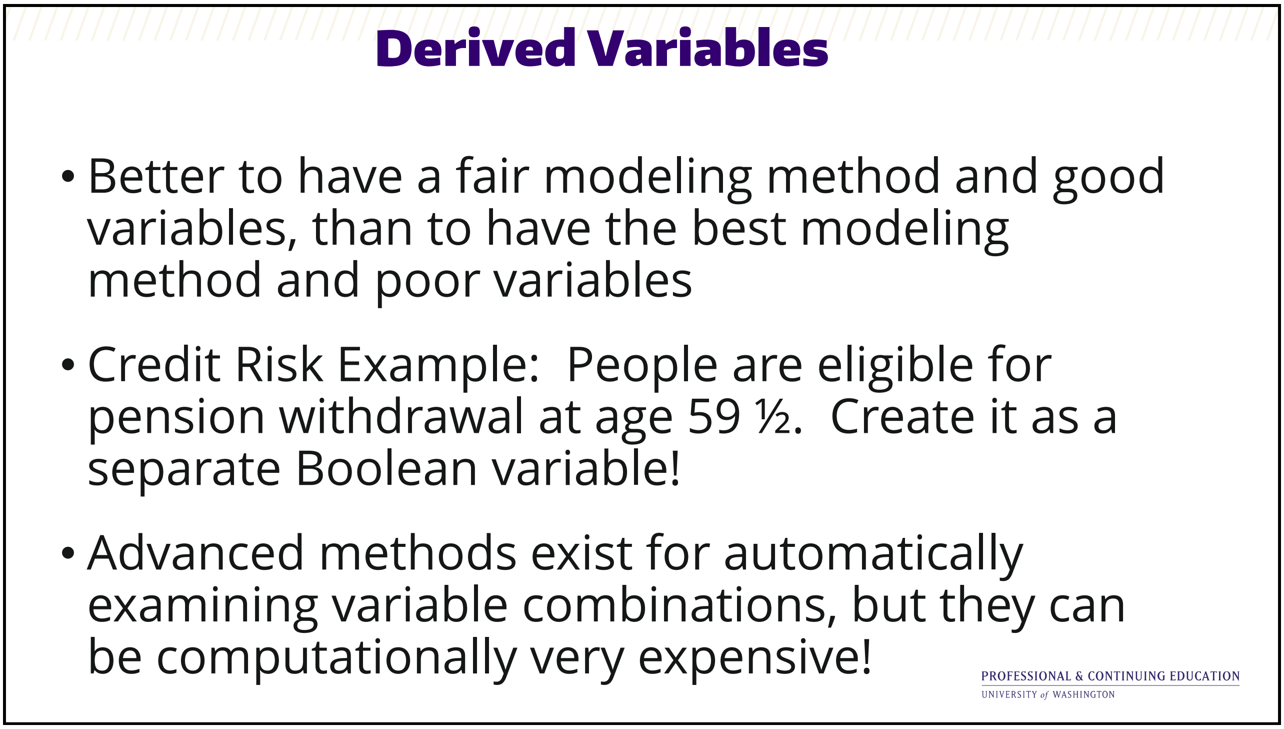
Equal Height = 4, except for the last bin

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

Equal-height partitioning: advantages

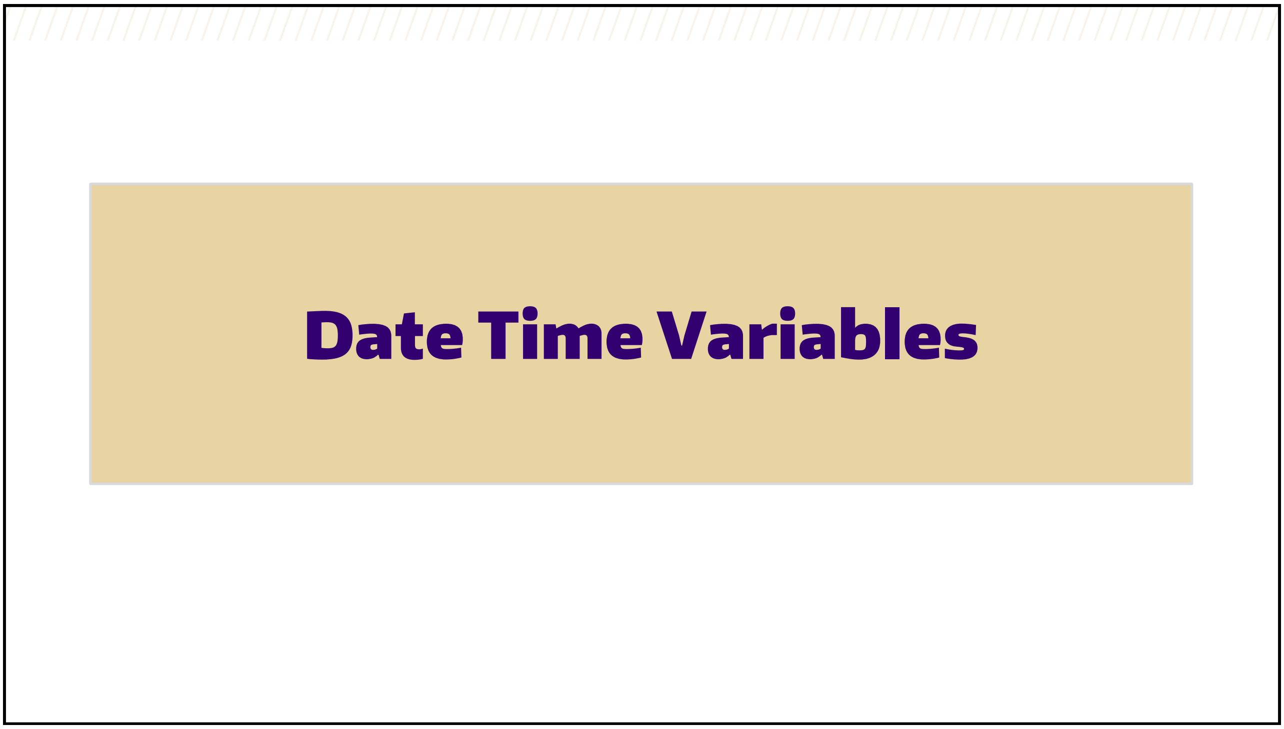
- Generally preferred because avoids clumping
- In practice, “almost-equal” height binning is used which avoids clumping and gives more intuitive breakpoints
- Additional considerations:
  - –don’t split frequent values across bins
  - –create separate bins for special values (e.g. 0)
  - –readable breakpoints (e.g. round breakpoints)

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON

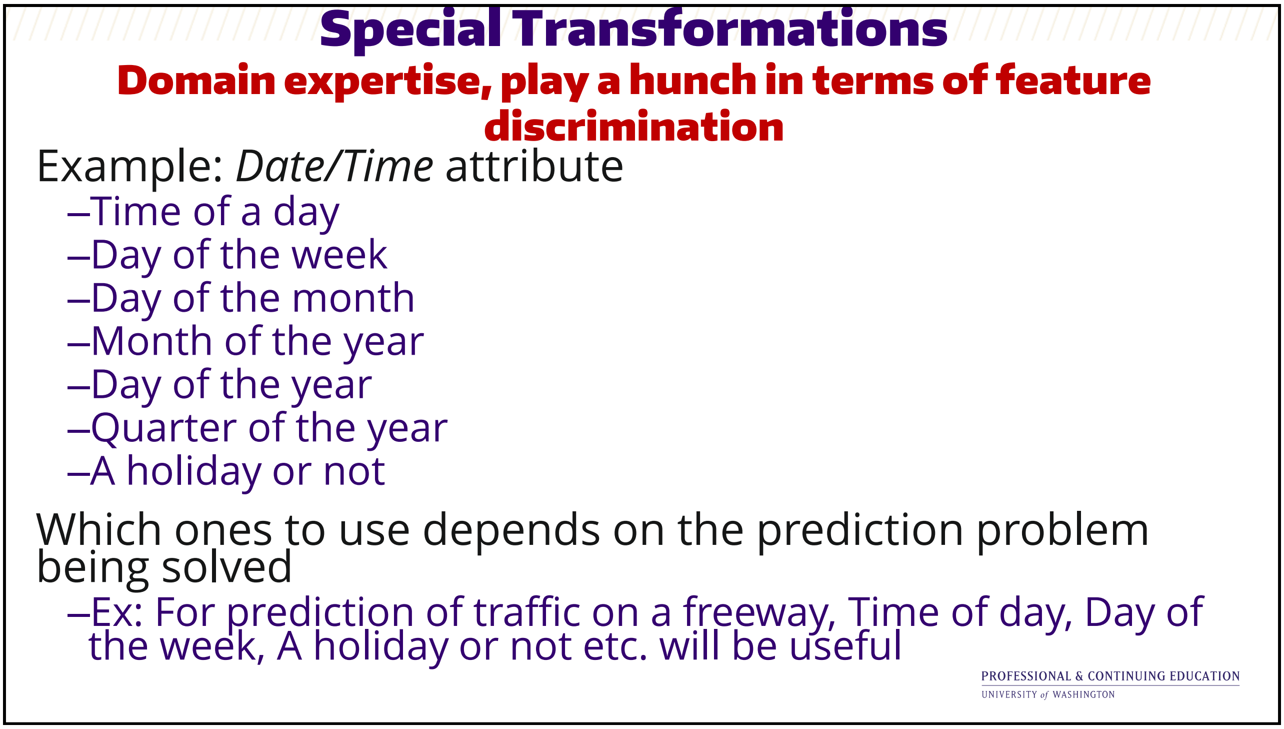


Derived Variables

- Better to have a fair modeling method and good variables, than to have the best modeling method and poor variables
- Credit Risk Example: People are eligible for pension withdrawal at age 59 ½. Create it as a separate Boolean variable!
- Advanced methods exist for automatically examining variable combinations, but they can be computationally very expensive!



Date Time Variables



Special Transformations

Domain expertise, play a hunch in terms of feature discrimination

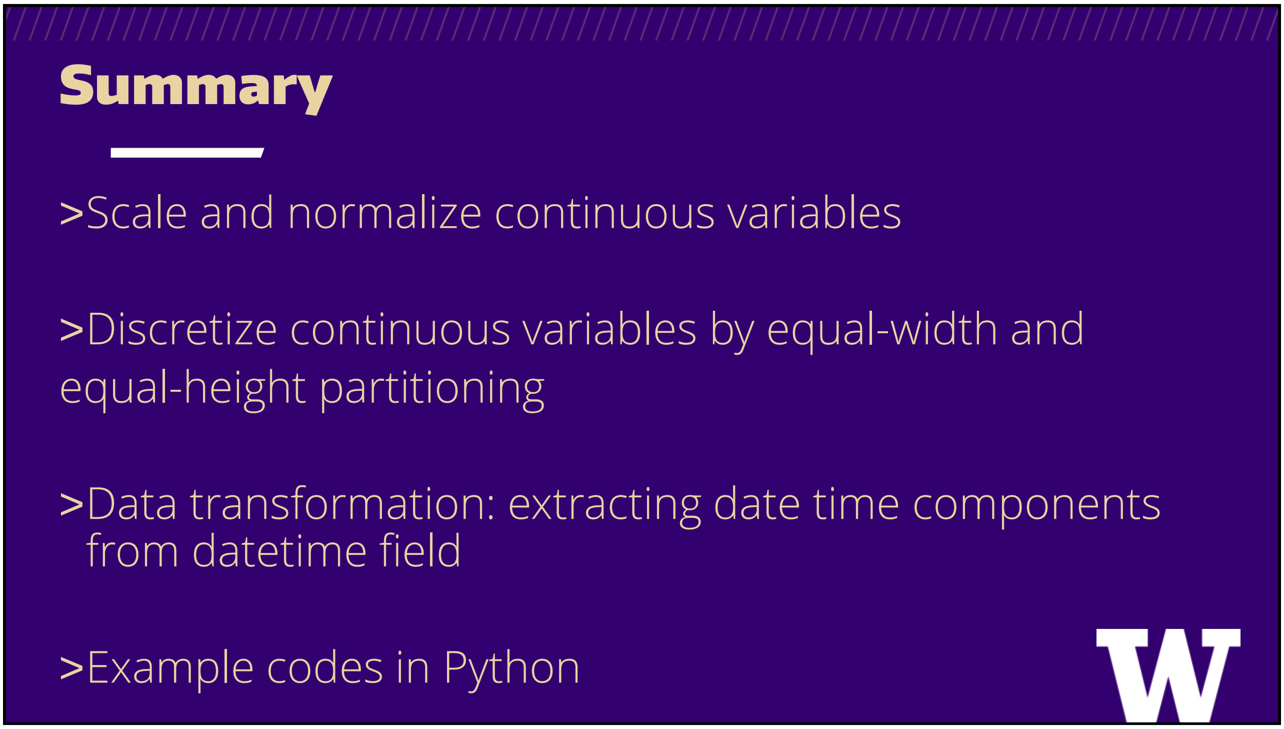
Example: *Date/Time* attribute

- –Time of a day
- –Day of the week
- –Day of the month
- –Month of the year
- –Day of the year
- –Quarter of the year
- –A holiday or not

Which ones to use depends on the prediction problem being solved

- –Ex: For prediction of traffic on a freeway, Time of day, Day of the week, A holiday or not etc. will be useful

PROFESSIONAL & CONTINUING EDUCATION
UNIVERSITY of WASHINGTON



Summary

>Scale and normalize continuous variables

>Discretize continuous variables by equal-width and equal-height partitioning

>Data transformation: extracting date time components from datetime field

>Example codes in Python