Analysis

INFO20002: Foundations of Informatics

- Outcomes:
  - Recognise different data formats
    - JSON
    - XML
    - CSV
    - Spreadsheet
  - Describe and select features that can answer “research questions/hypothesis”
  - Data process
    - Cleaning - error identification
    - Transforming, such as Aggregation / Mapping
  - Operations on Google sheet / EXCEL
    - Filter
    - Plotting

Recap on the knowledge

- Numerical variables (quantitative variables) refer to the numbers or anything that can have a range or can be measured along a continuum.
  - Temperature; Weight; height; distance
- Categorical variables (qualitative variables).
  - Nominal variables
    - Nominal variables have two or more categories
    - Nominal variables do not have an intrinsic order.
    - colours, nationalities, languages etc.
  - Ordinal variables
    - ordinal variables have two or more categories
    - ordinal variables can also be ordered or ranked
    - Year of birth : “2006”, “2007”, “2008”, “2009”

- Number variables can also be categorical, such as the ID, Codes and number-based options (e.g. “1-male” or “2-female”)

Exercise 1: Sourcing and understanding data

- 1. identify the formats of data sources
  - AFEVSD (CSV)
  - CDC-2011(JSON)
  - R/Tfrom1929 (Spreadsheet)
- 2. develop a research question/hypothesis
  - Are there more postal votes in a particular state?
  - Is diabetes more prevalent in lower income population?
  - Has rainfall increased since 1929 at Essendon Airport?
- 3. list the numerical and categorical variables
  - AFEVSD C: DivisionID; DivisionNm; StateAb/ N: all the others
  - CDC-2011 C: all
  - R/Tfrom1929 C: Product code; BoM station number; year; month; date; quality/ N: all the others
- Template

Types of errors

- **Semantic** – variables have similar meanings
  - "General Electric Company"
  - "General Elec Co."
  - "GE"
  - "Gen. Electric Company"
- **Range** – impossible values
  - Age = 150 years
  - Blood pressure = 250 mmHg
  - Percentage "108%"
- **Format errors**
  - Inconsistent with the "variable description"

Exercise 2: Data Cleaning

- 1. Identify the errors and error categories (Semantic; Range; Format)
  - A. B2, B3, B4 (semantic)
  - B. E8 (range) (over 100%)
  - C. E16 (format) (text instead of number)
  - D. A5 (range) (year in the future)
  - E. A12 (duplicates)

- 2. Save the new file with the name **“smoking-data-corrected.csv”**

Demonstratio

We can fix some parts manua
efficient and accurate to chec

The idea is checking the error
Standard.
Data published from the "
from the governments/ISO

- State.txt
- Load the "standard data" and "sample data"
- Check the variable based on these "principles"

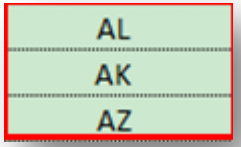
```
STATE|STUSAB|STATE_NAME|STATENS
01|AL|Alabama|01779775
02|AK|Alaska|01785533
04|AZ|Arizona|01779777
05|AR|Arkansas|00068085
06|CA|California|01779778
08|CO|Colorado|01779779
09|CT|Connecticut|01779780
10|DE|Delaware|01779781
11|DC|District of Columbia|01702382
12|FL|Florida|00294478
13|GA|Georgia|01705317
15|HI|Hawaii|01779782
16|ID|Idaho|01779783
17|IL|Illinois|01779784
18|IN|Indiana|00448508
19|IA|Iowa|01779785
20|KS|Kansas|00481813
21|KY|Kentucky|01779786
22|LA|Louisiana|01629543
23|ME|Maine|01779787
24|MD|Maryland|01714934
25|MA|Massachusetts|00606926
26|MI|Michigan|01779789
```

Error analysis: checking.py

	A	B	C	D	E	F
1	Year	State	Smoke everyday	Smoke some days	Former smoker	Never smoked
2	2010	AL	15.60%	6.30%	23.90%	54.20%
3	2010	AK	13.50%	6.80%	26.10%	53.60%
4	2010	AZ	10.70%	4.40%	27.90%	57.10%
5	2100	Arkansas	17.30%	5.60%	24.10%	53%
6	2010	California	7.50%	4.60%	23.10%	64.80%
7	2010	Colorado	11.40%	4.60%	24.70%	59.30%
8	2010	Connecticut	9.20%	4%	105.00%	57.60%
9	2010	Delaware	12.80%	4.50%	26.80%	56%
10	2010	District of Columbia	10%	5.70%	23.40%	61%
11	2010	Florida	12%	5.20%	29.80%	53%
12	2001	Georgia	12.80%	4.80%	23.10%	59.30%
13	2010	Guam	19.70%	6.10%	16.60%	57.60%
14	2010	Hawaii	10.70%	3.80%	25.30%	60.20%
15	2010	Idaho	11.30%	4.40%	22.90%	61.50%
16	2010	Illinois	11.50%	5.40%	nty-three point 6 perc	59.50%
17	2010	Indiana	16.30%	5%	25.10%	53.70%
18	2010	Iowa	12.10%	4.10%	23.40%	60.40%
19	2010	Kansas	11.90%	5.10%	24.20%	58.80%
20	2010	Kentucky	19.30%	5.50%	26%	49.20%

Semantic errors:

the state name and abbreviation – task 1



"AL": Alaska
"AK":

```
 #task 1 "for row in data:"
state = row['State'].strip() # The method strip() returns a copy of
if len(state) == 2: #Return the length of "string"
  if state in state_map:
    state = state_map[state]
    print 'Warning: two-letter state code is used for', year, state
  else:
    print 'Invalid state in the row for', year, state
    continue
```

Range error:

- Year: 1995 – 2010 – task 2

```
# task 2
if year > 2010 or year < 1995:
  print 'Year is out of range in the row for', year, state
  continue
```

- Percentage
  - [0-1] – task 3
  - Sum – task 4

```
measures = [
  'Smoke everyday',
  'Smoke some days',
  'Former smoker',
  'Never smoked'
]
```

```
  # task 3
values['State'] = state
total = 0
value = 0
# for every single percentage there are some rules need to follow
for measure in measures:
  value = check_percentage(row[measure])
  if value == -1:
    print 'Invalid percentage is found in the row for', year, state
    break
  # task 4
  values[measure] = value
  # the value is just the percetage number stripped the %
  total += value
if not total == 100: # additional checking
  #print 'Incorrect total is found in the row for', year, state
  pass
if value == -1:
  continue
```

```
def check_percentage(string):
  n = string.split('%')
  print n # the output is composed of two parts: e.g. ["50.70", ""]
  if len(n) != 2 or n[1] != '':
    return -1
  try:
    number = Decimal(n[0])
    if number > 100:
      return -1
    return number
  except:
    pass # do nothing
  return -1
```

Duplication:

"To a particular state, records for each year should only appear once" – task 5

Key is [state][year]

're-load' the data into an empty dictionary while checking

```
cube = defaultdict(dict)
data = csv.DictReader(open('smoking_data_us_1995_2010.csv'))
```

```
  # task 3
values['State'] = state
total = 0
value = 0
# for every single percentage there are some rules need to follow
for measure in measures:
  value = check_percentage(row[measure])
  if value == -1:
    print 'Invalid percentage is found in the row for', year, state
    break
  # task 4
  values[measure] = value
  # the value is just the percetage number stripped the %
  total += value
if not total == 100: # additional checking
  #print 'Incorrect total is found in the row for', year, state
  pass
if value == -1:
  continue
cube[state][year] = values
```

```
if year in cube[state]:
# 'cube' is initialised as an empty dictionary
#
#"in" and "if" sentence: the result is always depending on the part next to 'in'
  print 'Duplicate is found in the row for', year, state
  continue
```

Exercise 3: Data Transforming

- Based on the "**smoking-data-corrected.csv**":
  - 1. Mapping- Look at the column ***"year"*** ; map it to the year since start of data-sets and call it as ***"Indexed year"***
    - ***See the meaning of "indexed_year" on hands-on***
  - 2. Look at the column "Smoke everyday", "Smoke some days" and "Former smokers";

Method – transforming.py

- 1. Open and Initialize

```
fieldnames = ['Indexed Year', 'State', 'Never smoked', 'Has smoked']
csvout = open('smoking_data_us_1995_2010-transformed.csv', 'w')
writer = csv.DictWriter(csvout, fieldnames=fieldnames)
writer.writeheader()
```

- 2.1. ha

```
for row in data:
  values = {}
  year = int(row['Year'])
  indexed_year = year - 1995
  values['Indexed Year'] = indexed_year
```

xed_year”:

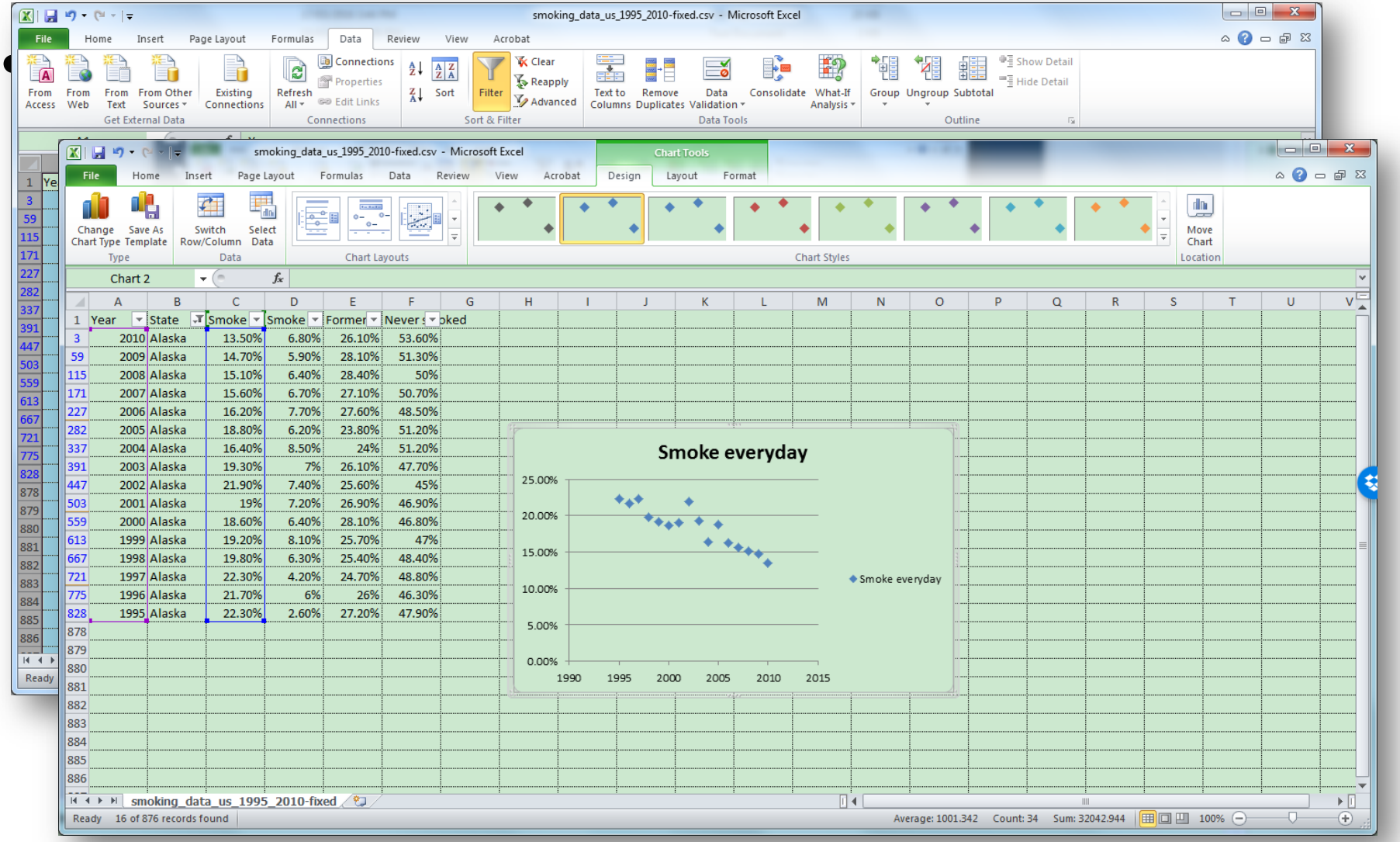
- measures = [ew

```
measures = [
  'Smoke everyday',
  'Smoke some days',
  'Former smoker',
  'Never smoked'
]
```

```
for measure in measures:
  # actually we have done this in the last work; the legal valu
  value = check_percentage(row[measure])
  # pass to the values
  values[measure] = value
  # do the aggregation
  #add a new column and calculate the number for this '1-'never
values['Has smoked'] = Decimal('100') - values['Never smoked']
#values as the dict of percentage numbers, again can be added i
cube[state][year] = values
#finally we can re-write the variables of the old into the new
writer.writerow({
  'Indexed Year': values['Indexed Year'],
  'State': values['State'],
  'Never smoked': values['Never smoked'],
  'Has smoked': values['Has smoked']
})
```

Part B Descriptive analysis

Feature selection



Conclude rules from datasets

- Try to characterize these trends into an association rule, such as…
  - *“if a person living in Alaska is asked about their smoking status, then they are XX more likely to answer with ‘never smoked’ than ‘smoke every day’ if questioned in 2010 than in 1995”*

Group formation

- Three students in each group (Ideally from the same workshop)
- How to choose datasets:
  - You can reference the websites/guidelines/questions abou dataset selection
  - The complexity and the level of sophistication of selected datasets is one part of assessment, so don't rush it.
- How to work with your partners?
- How to design the work?
  - Part C – discuss with your partners