Create a Tableau Story Project_PISA Data

By Shilin Li

Date: August 3rd, 2018

Data Wrangling

Gather

First, we download PISA data from Data Set Options (https://docs.google.com/document/d/1w7KhqotVi5eoKE3l_AZHbsxdr-NmcWsLTliZrpxWx4w/pub?embedded=true), then import it from local.

In [1]:

```
# import necessary library
import pandas as pd
```

In [2]:

```
# load only the first five rows of dataset to have a quick view
pd.read_csv('pisa2012.csv', nrows=5)
```

Out[2]:

	Unnamed: 0	CNT	SUBNATIO	STRATUM	OECD	NC	SCHOOLID	STIDSTD	ST
0	1	Albania	80000	ALB0006	Non-OECD	Albania	1	1	10
1	2	Albania	80000	ALB0006	Non-OECD	Albania	1	2	10
2	3	Albania	80000	ALB0006	Non-OECD	Albania	1	3	9
3	4	Albania	80000	ALB0006	Non-OECD	Albania	1	4	9
4	5	Albania	80000	ALB0006	Non-OECD	Albania	1	5	9

5 rows × 636 columns

In [3]:

```
# load the full dataset
df = pd.read_csv('pisa2012.csv', encoding='latin-1')
df.head()
```

```
/Users/shilinli/anaconda3/lib/python3.6/site-packages/IPython/core/i
nteractiveshell.py:2698: DtypeWarning: Columns (15,16,17,21,22,23,24
,25,26,30,31,36,37,45,65,123,155,156,157,158,159,160,161,162,163,164
,165,166,167,168,169,170,171,284,285,286,287,288,289,290,291,292,293
,294,295,296,297,298,299,300,301,302,303,307,308,309,310,311,312,313
,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330
,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347
,348,349,350,351,352,353,354,355,356,357,376,377,378,379,380,381,382
,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399
,400,401,402,403,475) have mixed types. Specify dtype option on impo
rt or set low_memory=False.
  interactivity=interactivity, compiler=compiler, result=result)
```

Out[3]:

	Unnamed: 0	CNT	SUBNATIO	STRATUM	OECD	NC	SCHOOLID	STIDSTD	ST
0	1	Albania	80000	ALB0006	Non-OECD	Albania	1	1	10
1	2	Albania	80000	ALB0006	Non-OECD	Albania	1	2	10
2	3	Albania	80000	ALB0006	Non-OECD	Albania	1	3	9
3	4	Albania	80000	ALB0006	Non-OECD	Albania	1	4	9
4	5	Albania	80000	ALB0006	Non-OECD	Albania	1	5	9

5 rows × 636 columns

In [4]:

```
# subset the dataset for those columns needed for later analysis in tableau
df1 = df[['CNT', 'STIDSTD', 'ST04Q01', 'ST27Q01', 'ST27Q02', 'ST27Q03', 'PV1MATH
','PV2MATH', 'PV3MATH', 'PV4MATH', 'PV5MATH', 'PV1READ', 'PV2READ',\
         'PV3READ', 'PV4READ', 'PV5READ', 'PV1SCIE', 'PV2SCIE', 'PV3SCIE', 'PV4S
CIE', 'PV5SCIE', 'ST49Q07', 'WEALTH']]
df1.head()
```

Out[4]:

	CNT	STIDSTD	ST04Q01	ST27Q01	ST27Q02	ST27Q03	PV1MATH	PV2MATH	PV
0	Albania	1	Female	Two	One	None	406.8469	376.4683	34
1	Albania	2	Female	Three or more	Three or more	Three or more	486.1427	464.3325	45
2	Albania	3	Female	Three or more	Two	Two	533.2684	481.0796	48
3	Albania	4	Female	Three or more	Two	One	412.2215	498.6836	41
4	Albania	5	Female	Two	One	Two	381.9209	328.1742	40

5 rows × 23 columns

Assess

Access is the second step, we will access them visually and programmatically, then recording any quality and tidiness issues found. Those issues will be resolved in the third step, cleaning.

In [5]:

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 485490 entries, 0 to 485489
Data columns (total 23 columns):
CNT         485490 non-null object
STIDSTD     485490 non-null int64
ST04Q01     485490 non-null object
ST27Q01     477079 non-null object
ST27Q02     476548 non-null object
ST27Q03     473459 non-null object
PV1MATH     485490 non-null float64
PV2MATH     485490 non-null float64
PV3MATH     485490 non-null float64
PV4MATH     485490 non-null float64
PV5MATH     485490 non-null float64
PV1READ     485490 non-null float64
PV2READ     485490 non-null float64
PV3READ     485490 non-null float64
PV4READ     485490 non-null float64
PV5READ     485490 non-null float64
PV1SCIE     485490 non-null float64
PV2SCIE     485490 non-null float64
PV3SCIE     485490 non-null float64
PV4SCIE     485490 non-null float64
PV5SCIE     485490 non-null float64
ST49Q07     312425 non-null object
WEALTH      479597 non-null float64
dtypes: float64(16), int64(1), object(6)
memory usage: 85.2+ MB
```

In [6]:

```
for item in list(df1):
    print(df1[item].unique())
```

```
['Albania' 'United Arab Emirates' 'Argentina' 'Australia' 'Austria'
 'Belgium' 'Bulgaria' 'Brazil' 'Canada' 'Switzerland' 'Chile' 'Colom
bia'
 'Costa Rica' 'Czech Republic' 'Germany' 'Denmark' 'Spain' 'Estonia'
 'Finland' 'France' 'United Kingdom' 'Greece' 'Hong Kong-China' 'Cro
atia'
 'Hungary' 'Indonesia' 'Ireland' 'Iceland' 'Israel' 'Italy' 'Jordan'
 'Japan' 'Kazakhstan' 'Korea' 'Liechtenstein' 'Lithuania' 'Luxembour
g'
 'Latvia' 'Macao-China' 'Mexico' 'Montenegro' 'Malaysia' 'Netherland
s'
 'Norway' 'New Zealand' 'Peru' 'Poland' 'Portugal' 'Qatar' 'China-Sh
anghai'
 'Perm(Russian Federation)' 'Florida (USA)' 'Connecticut (USA)'
```

```
 'Massachusetts (USA)' 'Romania' 'Russian Federation' 'Singapore' 'S
erbia'
 'Slovak Republic' 'Slovenia' 'Sweden' 'Chinese Taipei' 'Thailand'
 'Tunisia' 'Turkey' 'Uruguay' 'United States of America' 'Vietnam']
[    1     2     3 ..., 33804 33805 33806]
['Female' 'Male']
['Two' 'Three or more' 'One' nan 'None']
['One' 'Three or more' 'Two' nan 'None']
['None' 'Three or more' 'Two' 'One' nan]
[ 406.8469  486.1427  533.2684 ...,  178.4624  169.8941  824.7468]
[ 376.4683  464.3325  481.0796 ...,  173.9445  759.8613  758.7708]
[ 344.5319  453.4273  489.6479 ...,  787.5915  807.3765  808.3892]
[ 321.1637  472.9008  490.4269 ...,  128.7662  190.458   771.7791]
[ 381.9209  476.0165  533.2684 ...,  828.9531  838.3783  761.6529]
[ 249.5762  406.2936  401.21   ...,  693.7544  612.0262  672.9741]
[ 254.342   349.8975  404.3872 ...,  650.7822  653.2462  737.4415]
[ 406.8496  400.7334  387.7067 ...,  693.7544  654.7538  597.1101]
[ 175.7053  369.7553  431.3938 ...,  674.5779  623.7757  721.7323]
[ 218.5981  396.7618  401.21   ...,  715.2008  625.9801  685.0032]
[ 341.7009  548.9929  499.6643 ...,  715.5352  690.3581  691.2906]
[ 408.84    471.5964  428.7952 ...,  715.5352  690.3581  698.7505]
[ 348.2283  471.5964  492.2044 ...,  679.1682  687.5606  782.0216]
[ 367.8105  443.6218  512.7191 ...,  837.1317  714.6028  701.5479]
[ 392.9877  454.8116  499.6643 ...,  718.3327  696.8855  697.818 ]
['Never or rarely' nan 'Sometimes' 'Often' 'Always or almost always'
]
[-2.92  0.69 -0.23 -1.17 -0.95 -1.46 -0.49 -1.82 -0.73 -2.23 -2.59 -
1.6
 -1.74  0.34 -2.3  -2.05   nan -1.39  1.71 -0.18  0.04 -0.82 -1.7  -
2.27
 -3.34 -0.24 -1.16 -0.66  0.16  0.68 -1.41  0.31 -1.94 -3.92 -1.86
0.67
 -0.71 -1.26  2.85 -0.11 -0.93 -0.58  0.   -0.21  0.29 -1.42 -3.04
2.8
 -0.53  1.12 -1.01 -2.08 -2.78 -1.81 -2.31 -0.88 -1.67 -5.03  0.19 -
0.4  -1.
 -0.99 -0.59 -2.1  -2.28 -0.92 -2.25 -0.43 -0.98 -0.1  -2.26 -1.49 -
1.38
  1.16 -2.18 -1.91 -0.81 -1.09 -0.61 -1.92 -1.15 -1.32 -2.17 -1.55
0.2
 -1.51 -1.75 -1.97 -0.74 -2.71 -0.26 -1.72 -2.85 -2.98 -0.77  0.28 -
2.51
 -1.83 -1.99 -1.64 -0.2  -1.77 -3.03 -2.12 -0.91 -0.28 -1.43  0.23
0.01
 -1.35 -0.34 -1.05 -1.08 -1.59 -1.06 -2.82 -0.94 -0.27 -1.22 -2.36 -
0.86
 -3.26  0.59 -2.16  0.61 -3.24 -1.18 -1.45 -1.3  -0.55 -4.09 -2.48 -
1.88
 -2.57 -1.89 -0.76 -2.45 -1.03 -2.68 -1.25  0.12 -2.38 -2.72 -2.88 -
0.6
 -0.39  0.32 -0.09 -0.36 -1.61 -0.01 -1.87 -2.61 -1.69 -1.66 -0.37 -
0.67
 -1.53 -1.76 -3.28 -0.8  -1.28  0.43 -0.64 -1.19 -2.89 -1.24 -0.38 -
```

```
0.83
 -1.8   1.66 -0.13 -0.89 -3.09 -1.1  -3.16 -1.34 -1.47 -2.24 -2.73 -
4.59
 -1.63 -1.5   0.14 -0.08 -2.19 -1.11  2.77 -3.57 -1.62 -0.68 -2.34 -
2.21
 -0.47 -1.93  0.38 -2.49  0.48 -0.51 -4.51 -1.07 -1.54 -3.3  -0.65 -
1.44
 -2.02 -1.04 -0.56  0.65 -2.83 -0.3  -0.78  0.41 -1.48 -1.2  -2.79 -
1.33
 -3.45 -1.4  -2.14  1.23 -0.29 -0.54 -0.63  0.25 -0.9  -2.47 -0.04 -
2.56
 -2.01 -1.36 -0.44  0.97 -0.75 -2.13 -2.63  1.11 -0.5   0.63 -2.06 -
2.55
 -2.43 -1.27 -2.11 -3.62 -2.04 -1.65 -1.21  1.65  0.91 -2.22 -1.23 -
0.84
 -1.96 -1.79 -1.84 -4.68 -0.35  1.08 -3.21 -0.06 -0.48 -0.02 -4.88 -
2.07
  0.76 -2.2   0.98 -0.25 -0.14  2.76 -1.29  0.11  3.09  1.    0.37 -
1.31
  1.98  0.49  1.04  1.99  1.42 -0.46 -1.73  0.54  3.    3.03  1.88 -
1.58
  0.72  1.26  0.88 -1.68  0.36 -0.79 -0.33  0.5   0.07 -0.52  1.29
3.07
  2.97  0.58 -0.16 -0.05  0.09  0.42  0.51 -1.13  3.08 -0.41  0.84 -
1.85
 -4.77  2.15  0.52 -0.72  0.99  1.41  1.96 -0.57  1.81  1.61  1.24 -
0.07
 -3.63  0.86  1.75  2.84  0.77  1.6   2.53 -0.17  0.96  1.84 -0.22
1.85
  0.83 -2.46  1.38  1.43  1.01  0.93  3.06  0.45  1.55  1.62  1.67
1.3
  2.73  1.45  0.22  2.54  0.87  1.4  -0.69  0.85  2.81  2.96  1.59
2.12
  1.78  3.05  1.17  1.73  0.27 -1.56  1.83  2.17  0.46 -0.62  2.89
0.05
 -4.38  1.95  2.68  1.37  0.03  0.7  -0.15  2.79  1.93  0.02 -0.31
0.94
 -0.03  0.81  0.66 -3.47 -2.86  2.67  2.98 -3.29 -1.95  2.08  1.05
0.53
  0.08  0.35  1.33  0.6   0.26  2.83  0.73  1.18  1.86  1.9   1.1
1.27
  1.36  0.71  1.77  1.21  0.18  1.28  2.26 -0.42 -1.02  0.89 -2.33 -
0.19
  2.82  0.33  2.99  1.69  0.57  1.39 -1.12  1.06 -3.54  1.72  1.19 -
2.69
 -1.14 -2.09 -0.12 -3.07 -2.37  0.62 -1.37 -3.46 -1.57 -1.98 -3.52
0.3
 -5.35 -1.52 -3.   -0.87 -2.96 -4.18 -2.81 -0.96  0.24 -2.44  0.44 -
0.7
 -1.9  -1.78 -2.54 -2.77 -3.48 -0.32  0.17  1.14 -2.15 -2.99 -2.   -
2.6
 -2.58 -0.85  1.46 -2.65 -3.56 -3.02 -2.66 -2.91  2.03  1.92  3.12
2.02
```

```
  0.8   0.79  3.02 -2.03  0.13  1.35  2.22  1.74 -5.28  0.1   1.34
0.15
  1.91  0.95  0.06  1.8   1.87  0.78  1.58  2.62 -3.99 -3.19  1.22 -
2.64
  1.82  0.64  1.57 -2.74  1.25  0.56  1.07 -0.45 -2.42  1.79  1.13
3.04
  2.93  2.01  1.68  1.15 -3.66  1.94  1.76  2.44  3.01  0.39  2.87
0.75
  1.09 -5.12  2.59  0.21  1.7   1.03  1.64  0.4   1.02  0.74  1.53
2.13
 -0.97  0.82  1.97 -5.34  0.92 -3.32 -4.05 -3.13 -3.72 -4.85  2.94 -
2.5
 -2.84 -2.35 -4.16 -2.4   0.55 -2.41  1.2  -3.89 -4.65 -5.08 -3.91
2.36
  2.91 -3.27  2.4   2.43 -3.12 -2.97 -3.8  -4.35  2.9  -3.14 -3.23 -
2.67
 -4.21 -2.39 -3.01 -2.76 -1.71 -2.95 -3.59 -3.08 -3.06 -3.2  -3.5  -
3.53
 -2.7  -2.62 -5.07 -3.25 -3.88 -2.32 -2.29 -3.87 -3.15  2.55 -3.85 -
2.93
 -4.73 -4.5  -3.39 -4.64  1.49 -3.9  -2.94 -3.18  1.32  2.27 -3.84 -
3.61
 -2.8  -3.86  2.28 -4.6  -4.02 -3.41 -3.78 -4.03  2.41  1.31  1.48
2.86
  2.78 -5.48  2.58  2.63 -5.24  2.3   1.56  1.89 -5.11 -3.31  1.63 -
2.9
 -5.04 -3.33 -2.52 -2.53  1.5   2.29 -3.44 -5.45 -3.83 -3.1  -5.4
0.47
 -4.26 -4.36 -4.17 -3.42 -4.13 -5.63 -4.23 -5.42 -4.31 -3.17 -3.6  -
4.24
 -4.49 -3.35 -3.38 -4.46 -4.37 -5.47 -3.55 -3.82 -3.73 -4.44  0.9  -
3.36
 -5.43 -4.48 -3.77  2.88 -4.42 -3.81 -3.11 -5.17 -3.51 -5.32 -4.99
2.42
  2.64 -4.07 -5.01 -4.78  1.47  2.95  2.52  2.61 -4.15  2.92  3.13
2.
 -4.12 -3.43 -3.4  -4.34 -2.75 -3.67 -3.74 -5.53 -3.64 -3.65 -5.33 -
5.27
 -5.52 -3.05 -4.   -4.3  -4.32 -3.96 -4.29 -5.3  -4.01 -3.98 -2.87 -
5.31
  2.14  3.25 -4.19  2.74  2.38  1.44 -4.7  -4.8  -3.7  -4.58  1.51 -
4.67
 -3.94  2.47 -5.2   2.39 -3.79  3.11  2.25 -4.96 -4.04 -5.19 -3.37 -
5.13
 -5.1  -4.84 -4.83 -4.52 -3.22 -3.76 -3.97 -3.49 -3.71 -4.81 -4.97 -
5.16
 -4.92 -4.72 -5.06  2.51 -3.93 -5.15 -4.25 -5.14  1.54 -3.95 -6.65
3.16
  3.15  2.35 -5.26 -4.93 -4.71 -5.09 -5.36 -4.14 -3.75  2.18 -4.11 -
4.56
 -4.86  2.66 -3.58  2.16  2.75  2.31  2.19  2.45  2.37  2.6  -4.75
2.46
  2.69  2.49 -3.69 -4.62 -4.89  3.18 -4.66  2.33 -4.87 -5.   -4.27 -
```

```
4.79
 -6.08 -6.   -4.55 -4.53 -4.2 ]
```

Quality

- Column names are ambiguous;
- Using plausible values to obtain score;
- Adding gender count columns

Tidiness

- Drop useless columns
- Melt apparatus columns;

Save the new dataframe

Clean

In [7]:

```
# make a copy
df_clean = df1.copy()
```

In [8]:

```
# rename the column names
df_clean.rename(columns={
                'CNT': 'country',
                'STIDSTD': 'student_id',
                'ST04Q01': 'gender',
                'ST27Q01': 'cellular_phone',
                'ST27Q02': 'TV',
                'ST27Q03': 'computers',
                'ST49Q07': 'computer_programming',
                'WEALTH': 'family_wealth_index'
                }, inplace = True)
list(df_clean)
```

Out[8]:

```
['country',
 'student_id',
 'gender',
 'cellular_phone',
 'TV',
 'computers',
 'PV1MATH',
 'PV2MATH',
 'PV3MATH',
 'PV4MATH',
 'PV5MATH',
 'PV1READ',
 'PV2READ',
 'PV3READ',
 'PV4READ',
 'PV5READ',
 'PV1SCIE',
 'PV2SCIE',
 'PV3SCIE',
 'PV4SCIE',
 'PV5SCIE',
 'computer_programming',
 'family_wealth_index']
```

In [9]:

```
# calculate the math, read and science score
df_clean['math_score'], df_clean['reading_score'], df_clean['science_score'] = \
df_clean[['PV1MATH', 'PV2MATH', 'PV3MATH', 'PV4MATH', 'PV5MATH']].mean(axis=1),
\
df_clean[['PV1READ', 'PV2READ', 'PV3READ', 'PV4READ', 'PV5READ']].mean(axis=1),\
df_clean[['PV1SCIE', 'PV2SCIE', 'PV3SCIE', 'PV4SCIE', 'PV5SCIE']].mean(axis=1)

df_clean.head()
```

Out[9]:

	country	student_id	gender	cellular_phone	TV	computers	PV1MATH	PV2MATH
0	Albania	1	Female	Two	One	None	406.8469	376.4683
1	Albania	2	Female	Three or more	Three or more	Three or more	486.1427	464.3325
2	Albania	3	Female	Three or more	Two	Two	533.2684	481.0796
3	Albania	4	Female	Three or more	Two	One	412.2215	498.6836
4	Albania	5	Female	Two	One	Two	381.9209	328.1742

5 rows × 26 columns

In [10]:

```
# drop useless columns
df_tableau = df_clean[['country','student_id', 'gender', 'cellular_phone', 'TV',
'computers',\
                       'computer_programming', 'math_score', 'reading_score', 'sc
ience_score', 'family_wealth_index']]
```

In [11]:

```
# randomly check the new dataframe
df_tableau.sample(5)
```

Out[11]:

	country	student_id	gender	cellular_phone	TV	computers	computer_pro
173525	Estonia	2630	Female	Three or more	One	Two	Never or rarely
220775	Hungary	4197	Male	Two	Two	One	NaN
336879	Mexico	21177	Male	Two	Two	None	Often
184257	Finland	8583	Female	Three or more	None	One	Often
133172	Germany	72	Female	Three or more	Two	Three or more	Never or rarely

In [12]:

```
# melt apparatus columns
df_tableau = pd.melt(df_tableau, id_vars=['country', 'student_id', 'gender', 'co
mputer_programming', 'math_score', \
                                'reading_score', 'science_score', 'family_wealth_ind
ex'], value_vars=['cellular_phone',\

'TV',\

'computers'])
# rename the new columns
df_tableau.rename(columns={
                'variable': 'electronics',
                'value': 'electronic_numbers'
                }, inplace = True)

df_tableau.sample(5)
```

Out[12]:

	country	student_id	gender	computer_programming	math_score	reading_
1335141	Norway	252	Male	Often	467.83768	423.4889
531931	Belgium	5055	Male	NaN	527.03692	428.6214
51999	Bulgaria	2016	Female	Never or rarely	558.50600	618.7715
554477	Brazil	13722	Male	NaN	348.11496	378.8205
680234	United Kingdom	5628	Female	Never or rarely	608.43590	565.3144

In [13]:

```
# add gender count columns
male_count = []
female_count = []

for item in df_tableau.gender:
    if item == 'Male':
        male_count.append(1)
        female_count.append(0)
    else:
        male_count.append(0)
        female_count.append(1)

df_tableau['male_count'] = male_count
df_tableau['female_count'] = female_count

df_tableau.sample(5)
```

Out[13]:

	country	student_id	gender	computer_programming	math_score	reading_s
308430	Latvia	2369	Female	NaN	367.27684	375.3949(
275955	Jordan	4293	Male	Always or almost always	381.37558	364.06482
1124961	Spain	8399	Female	NaN	502.34462	531.3180(
932377	Sweden	3293	Male	Often	446.96214	433.11228
796096	Macao-China	239	Male	Never or rarely	503.82458	462.38324

Save the new dataframe

In [14]:

```
# save the new dataframe to local
df_tableau.to_csv('pisa_tableau_2012.csv', encoding='utf-8', index=False)
```

In [15]:

```
from subprocess import call
call(['python', '-m', 'nbconvert', 'Wrangle_Act_Create a Tableau Story Project.ipynb'])
```

Out[15]:

```
0
```