Day 2, Part 5: Linear Regression & Communicating Results

Introduction to Python

Tom Paskhalis

RECSM Summer School 2023

Anscombe's quartet

- 4 artificial datasets constructed by Anscombe (1973)
- All of them have nearly identical summary statistics
- But show dramatically different relationships between variables
- Designed to illustrate the importance of data visualization

Data for Anscombe's quartet

Data for Anscombe’s quartet

In [1]:

```
import pandas as pd
anscombe_quartet = pd.read_csv('../data/anscombes_quartet.csv')
```

Data for Anscombe's quartet

In [1]:
```
import pandas as pd
anscombe_quartet = pd.read_csv('../data/anscombes_quartet.csv')
```

In [2]:
```
anscombe_quartet.head()
```

Out[2]:

	dataset	x	y
0	I	10	8.04
1	I	8	6.95
2	I	13	7.58
3	I	9	8.81
4	I	11	8.33

Summary statistics for Anscombe's quartet

Summary statistics for Anscombe's quartet

In [3]:

```
# Here we use `groupby` method to create summary by a variable ('datase
anscombe_quartet.groupby(['dataset']).describe()
```

Out[3]:

	x									
	count	mean	std	min	25%	50%	75%	max	count	mean
dataset										
I	11.0	9.0	3.316625	4.0	6.5	9.0	11.5	14.0	11.0	7.500909
II	11.0	9.0	3.316625	4.0	6.5	9.0	11.5	14.0	11.0	7.500909
III	11.0	9.0	3.316625	4.0	6.5	9.0	11.5	14.0	11.0	7.500000
IV	11.0	9.0	3.316625	8.0	8.0	8.0	8.0	19.0	11.0	7.500909

Plotting Anscombe's quartet

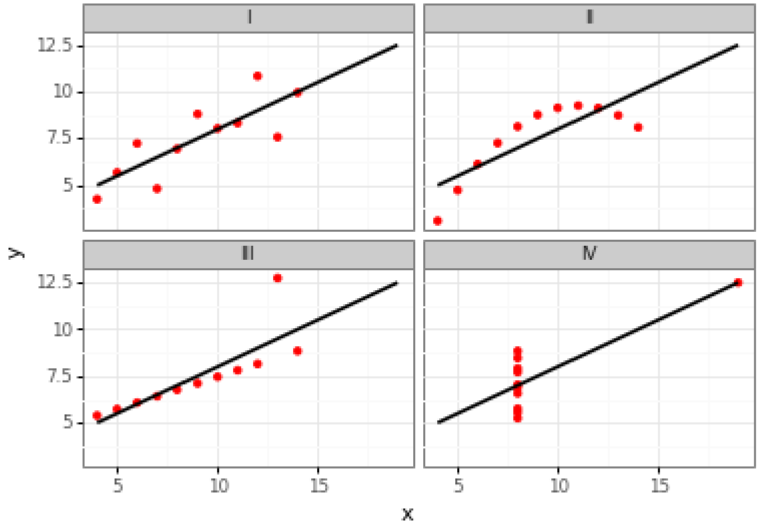
Plotting Anscombe's quartet

In [4]:

```
from plotnine import *

ggplot(anscombe_quartet, aes(x = 'x', y = 'y')) +\
    geom_point(colour = 'red') +\
    geom_smooth(method = 'lm', se = False, fullrange = True) +\
    facet_wrap('dataset') +\
    theme_bw()
```

```
/home/tpaskhalis/Decrypted/Git/RECSM/venv/lib/python3.8/site-pac
kages/plotnine/utils.py:371: FutureWarning: The frame.append met
hod is deprecated and will be removed from pandas in a future ve
rsion. Use pandas.concat instead.
/home/tpaskhalis/Decrypted/Git/RECSM/venv/lib/python3.8/site-pac
kages/plotnine/utils.py:371: FutureWarning: The frame.append met
hod is deprecated and will be removed from pandas in a future ve
rsion. Use pandas.concat instead.
```



Out[4]: <ggplot: (8774988838066)>

Linear regression

- Linear regression is the classical tool of statistical analysis
- It allows to estimate the degree of association between variables
- Typically, it is the association between one or more independent variables (IV) and one dependent variable (DV)
- The main quantities of interest usually are direction, magnitude of association and its statistical significance

Linear regression in Python

- As for tabular data and visualization we need external libraries for running regression
- `statsmodels` library provides tools for estimating many statistical models
- Another useful library is `scikit-learn`
- It is more focussed on machine-learning applications

Linear regression in Python

- As for tabular data and visualization we need external libraries for running regression
- `statsmodels` library provides tools for estimating many statistical models
- Another useful library is `scikit-learn`
- It is more focussed on machine-learning applications

In [5]:
```
import statsmodels.api as sm
import statsmodels.formula.api as smf # Formula API provides R-style fo
```

Data transformation

Data transformation

In [6]:
```
kaggle2021 = pd.read_csv('../data/kaggle_survey_2021_responses.csv', sk
```

```
/tmp/ipykernel_24423/2749400638.py:1: DtypeWarning: Columns (19
5,201) have mixed types. Specify dtype option on import or set l
ow_memory=False.
```

Data transformation

In [6]:

```
kaggle2021 = pd.read_csv('../data/kaggle_survey_2021_responses.csv', sk
```

```
/tmp/ipykernel_24423/2749400638.py:1: DtypeWarning: Columns (19
5,201) have mixed types. Specify dtype option on import or set l
ow_memory=False.
```

In [7]:

```
# Let's give more intuitive names to out variables
kaggle2021 = kaggle2021.rename(columns = {
                'Q1': 'age',
                'Q2': 'gender',
                'Q3': 'country',
                'Q4': 'education',
                'Q25': 'compensation'})
```

Data transformation

In [6]:

```
kaggle2021 = pd.read_csv('../data/kaggle_survey_2021_responses.csv', sk
```

```
/tmp/ipykernel_24423/2749400638.py:1: DtypeWarning: Columns (19
5,201) have mixed types. Specify dtype option on import or set l
ow_memory=False.
```

In [7]:

```
# Let's give more intuitive names to out variables
kaggle2021 = kaggle2021.rename(columns = {
                'Q1': 'age',
                'Q2': 'gender',
                'Q3': 'country',
                'Q4': 'education',
                'Q25': 'compensation'})
```

In [8]:

```
kaggle2021['compensation'].head(n = 2)
```

Out[8]:

```
0    25,000-29,999
1    60,000-69,999
Name: compensation, dtype: object
```

Data transformation

In [6]:

```
kaggle2021 = pd.read_csv('../data/kaggle_survey_2021_responses.csv', sk
```

```
/tmp/ipykernel_24423/2749400638.py:1: DtypeWarning: Columns (19
5,201) have mixed types. Specify dtype option on import or set l
ow_memory=False.
```

In [7]:

```
# Let's give more intuitive names to out variables
kaggle2021 = kaggle2021.rename(columns = {
                'Q1': 'age',
                'Q2': 'gender',
                'Q3': 'country',
                'Q4': 'education',
                'Q25': 'compensation'})
```

In [8]:

```
kaggle2021['compensation'].head(n = 2)
```

Out[8]:

```
0    25,000-29,999
1    60,000-69,999
Name: compensation, dtype: object
```

In [9]:

```
from statistics import mean
# Here we are replacing the compensation range by its midpoint (i.e. 1:
# This variable requires substantial cleaning before transformation
# Such as extraneous symbols ('$', ',', '>') have to be removed
```

```
kaggle2021['compensation'] = kaggle2021['compensation'].map(
    lambda x: mean([float(x.replace(',','').replace('$','').replace('>'
)
```

Pandas and linear regression

Pandas and linear regression

In [10]:

```
kaggle2021['compensation'] # Level of compensation (in USD) - our DV
```

Out[10]:

```
0        27499.5
1        64999.5
2          499.5
3        34999.5
4        34999.5
          ...
25968    17499.5
25969        NaN
25970      499.5
25971        NaN
25972      499.5
Name: compensation, Length: 25973, dtype: float64
```

Pandas and linear regression

In [10]:
```
kaggle2021['compensation'] # Level of compensation (in USD) - our DV
```

Out[10]:
```
0        27499.5
1        64999.5
2          499.5
3        34999.5
4        34999.5
          ...
25968    17499.5
25969        NaN
25970      499.5
25971        NaN
25972      499.5
Name: compensation, Length: 25973, dtype: float64
```

In [11]:
```
kaggle2021['gender'].value_counts() # Frequencies of gender categories
```

Out[11]:
```
Man                          20598
Woman                         4890
Prefer not to say              355
Nonbinary                       88
Prefer to self-describe         42
Name: gender, dtype: int64
```

Pandas and linear regression

In [10]:
```
kaggle2021['compensation'] # Level of compensation (in USD) - our DV
```

Out[10]:
```
0        27499.5
1        64999.5
2          499.5
3        34999.5
4        34999.5
           ...
25968    17499.5
25969        NaN
25970      499.5
25971        NaN
25972      499.5
Name: compensation, Length: 25973, dtype: float64
```

In [11]:
```
kaggle2021['gender'].value_counts() # Frequencies of gender categories
```

Out[11]:
```
Man                        20598
Woman                       4890
Prefer not to say            355
Nonbinary                     88
Prefer to self-describe       42
Name: gender, dtype: int64
```

In [12]:
```
# Formula specification allows to write 'DV ~ IV_1 + IV_2 + ... + IV_N'
fit1 = smf.ols('compensation ~ gender', data = kaggle2021).fit()
```

Model summary

Model summary

In [13]:
```
fit1.summary()
```

Out[13]:

OLS Regression Results

Dep. Variable:	compensation	**R-squared:**	0.007
Model:	OLS	**Adj. R-squared:**	0.007
Method:	Least Squares	**F-statistic:**	26.46
Date:	Mon, 27 Jun 2022	**Prob (F-statistic):**	6.64e-22
Time:	22:08:46	**Log-Likelihood:**	-1.9707e+05
No. Observations:	15391	**AIC:**	3.941e+05
Df Residuals:	15386	**BIC:**	3.942e+05
Df Model:	4		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	4.593e+04	782.805	58.672	0.000	4.44e+04	4.75e+04
gender[T.Nonbinary]	7.016e+04	1.29e+04	5.455	0.000	4.49e+04	9.54e+04
gender[T.Prefer not to say]	2.166e+04	6335.402	3.419	0.001	9242.046	3.41e+04
gender[T.Prefer to self-describe]	2.498e+04	1.8e+04	1.389	0.165	-1.03e+04	6.02e+04

gender[T.Woman]	-1.466e+04	1932.351	-7.588	0.000	-1.84e+04	-1.09e+04

Omnibus:	18458.954	**Durbin-Watson:**	2.001
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2343660.164
Skew:	6.472	**Prob(JB):**	0.00
Kurtosis:	62.051	**Cond. No.**	25.7

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Multiple linear regression

Multiple linear regression

In [14]:

```
# Let's now also control for age and education
fit2 = smf.ols('compensation ~ gender + age + education', data = kaggle
```

Multiple linear regression

In [14]:

```
# Let's now also control for age and education
fit2 = smf.ols('compensation ~ gender + age + education', data = kaggle
```

In [15]:

```
fit2.summary()
```

Out[15]:

OLS Regression Results

Dep. Variable:	compensation	**R-squared:**	0.068
Model:	OLS	**Adj. R-squared:**	0.067
Method:	Least Squares	**F-statistic:**	55.98
Date:	Mon, 27 Jun 2022	**Prob (F-statistic):**	1.77e-216
Time:	22:08:46	**Log-Likelihood:**	-1.9658e+05
No. Observations:	15391	**AIC:**	3.932e+05
Df Residuals:	15370	**BIC:**	3.934e+05
Df Model:	20		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.9**
Intercept	1.483e+04	2910.746	5.093	0.000	9120.353	2.05e-
gender[T.Nonbinary]	6.443e+04	1.25e+04	5.163	0.000	4e+04	8.89e-
gender[T.Prefer not to say]	2.033e+04	6157.104	3.302	0.001	8261.253	3.24e-

gender[T.Prefer to self-describe]	3.486e+04	1.74e+04	1.998	0.046	667.732	6.91e-
gender[T.Woman]	-1.1e+04	1884.889	-5.838	0.000	-1.47e+04	-7309.5
age[T.22-24]	2313.5562	3387.043	0.683	0.495	-4325.450	8952.5
age[T.25-29]	8811.2235	3235.996	2.723	0.006	2468.289	1.52e-
age[T.30-34]	2.427e+04	3347.112	7.251	0.000	1.77e+04	3.08e-
age[T.35-39]	3.405e+04	3485.133	9.770	0.000	2.72e+04	4.09e-
age[T.40-44]	4.192e+04	3656.035	11.465	0.000	3.48e+04	4.91e-
age[T.45-49]	5.368e+04	3872.345	13.862	0.000	4.61e+04	6.13e-
age[T.50-54]	5.354e+04	4211.610	12.713	0.000	4.53e+04	6.18e-
age[T.55-59]	6.666e+04	4792.714	13.910	0.000	5.73e+04	7.61e-
age[T.60-69]	5.718e+04	4971.283	11.502	0.000	4.74e+04	6.69e-
age[T.70+]	6.918e+04	9180.635	7.535	0.000	5.12e+04	8.72e-
education[T.Doctoral degree]	1.187e+04	2322.458	5.111	0.000	7318.325	1.64e-
education[T.I prefer not to answer]	-1.029e+04	4857.239	-2.118	0.034	-1.98e+04	-769.1
education[T.Master's degree]	7168.1892	1666.521	4.301	0.000	3901.611	1.04e-
education[T.No formal education past high school]	-1.059e+04	5801.400	-1.825	0.068	-2.2e+04	782.1
education[T.Professional doctorate]	8857.2561	5216.498	1.698	0.090	-1367.698	1.91e-

education[T.Some college/university study without earning a bachelor's degree]	-912.0023	3378.159	-0.270	0.787	-7533.593	5709.5

Omnibus:	19216.780	**Durbin-Watson:**	2.008
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	2958551.621
Skew:	6.892	**Prob(JB):**	0.00
Kurtosis:	69.509	**Cond. No.**	30.2

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Markdown - a language of reports

- Markdown is a markup language for formatting text with simple syntax
- The key goal of Markdown is readability
- Only a limited set of formatting options is supported
- Markdown is used in online documentation, blogging and instant messaging

Formatting text in Markdown

- For *italics* `*one star on each side*`
- For **bold** `**two stars on each side**`
- For ~~strikethrough~~ `~~two tildes on each side~~`

Lists in Markdown

For bulleted or unordered list of items:

```
- Just add a dash first and then write a text.
- If you add another dash in the following line, you will have another item in the list.
  - If you add four spaces or use a tab key, you will create an indented list.
```

For numbered or ordered list of items:

```
1. Just type a number and then write a text.
2. If you want to add a second item, just type in another number.
1. If you make a mistake when typing numbers, fear not, Markdown will correct it for you.
    1. If you press a tab key or type four spaces, you will get an indented list and the numbering
    will start from scratch.
```

Headers in Markdown

Headers or section titles are created with hashes(`#`)

`# This is a first-tier header`

`## This is a second-tier header`

`### This is a third-tier header`

Images and links in Markdown

- To add an image you can write `![some text](image_path)`
- To add a link you can write `[some text](URL)`
- For more complex cases HTML code can be used

Tables in Markdown

- Tables in Markdown can be created using the following syntax (there are a few variants)

```
| Header1 | Header2 |
|:--------|:--------|
| content | content |
```

- `:---` produces left-aligned text in cells
- `---:` produces right-aligned text in cells
- `:---:` produces centered text in cells

Markdown tables in pandas

- Pandas can generate Markdown tables from DataFrame

Markdown tables in pandas

- Pandas can generate Markdown tables from DataFrame

In [16]:

```
# Let's revisit the summary statistics of Anscombe's quartet
anscombe_quartet.groupby(['dataset']).describe().iloc[:,0:3]
```

Out[16]:

	x		
	count	**mean**	**std**
dataset			
I	11.0	9.0	3.316625
II	11.0	9.0	3.316625
III	11.0	9.0	3.316625
IV	11.0	9.0	3.316625

Markdown tables in pandas

- Pandas can generate Markdown tables from DataFrame

In [16]:

```
# Let's revisit the summary statistics of Anscombe's quartet
anscombe_quartet.groupby(['dataset']).describe().iloc[:,0:3]
```

Out[16]:

	x		
	count	**mean**	**std**
dataset			
I	11.0	9.0	3.316625
II	11.0	9.0	3.316625
III	11.0	9.0	3.316625
IV	11.0	9.0	3.316625

In [17]:

```
print(anscombe_quartet.groupby(['dataset']).describe().iloc[:,0:3].to_n
```

```
|   ('x', 'count') |   ('x', 'mean') |   ('x', 'std') |
|-----------------:|----------------:|---------------:|
|               11 |               9 |        3.31662 |
|               11 |               9 |        3.31662 |
|               11 |               9 |        3.31662 |
|               11 |               9 |        3.31662 |
```

The end