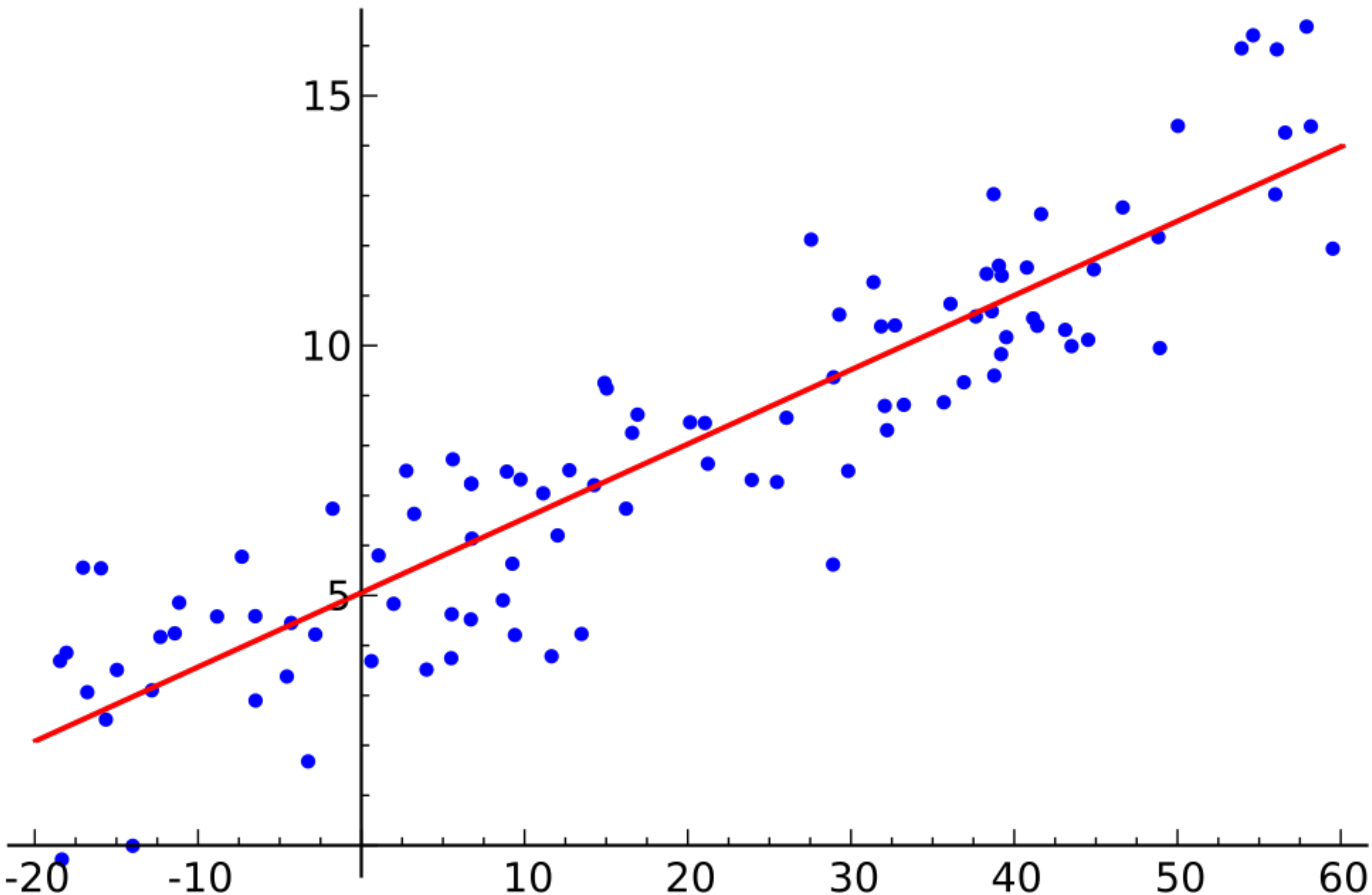
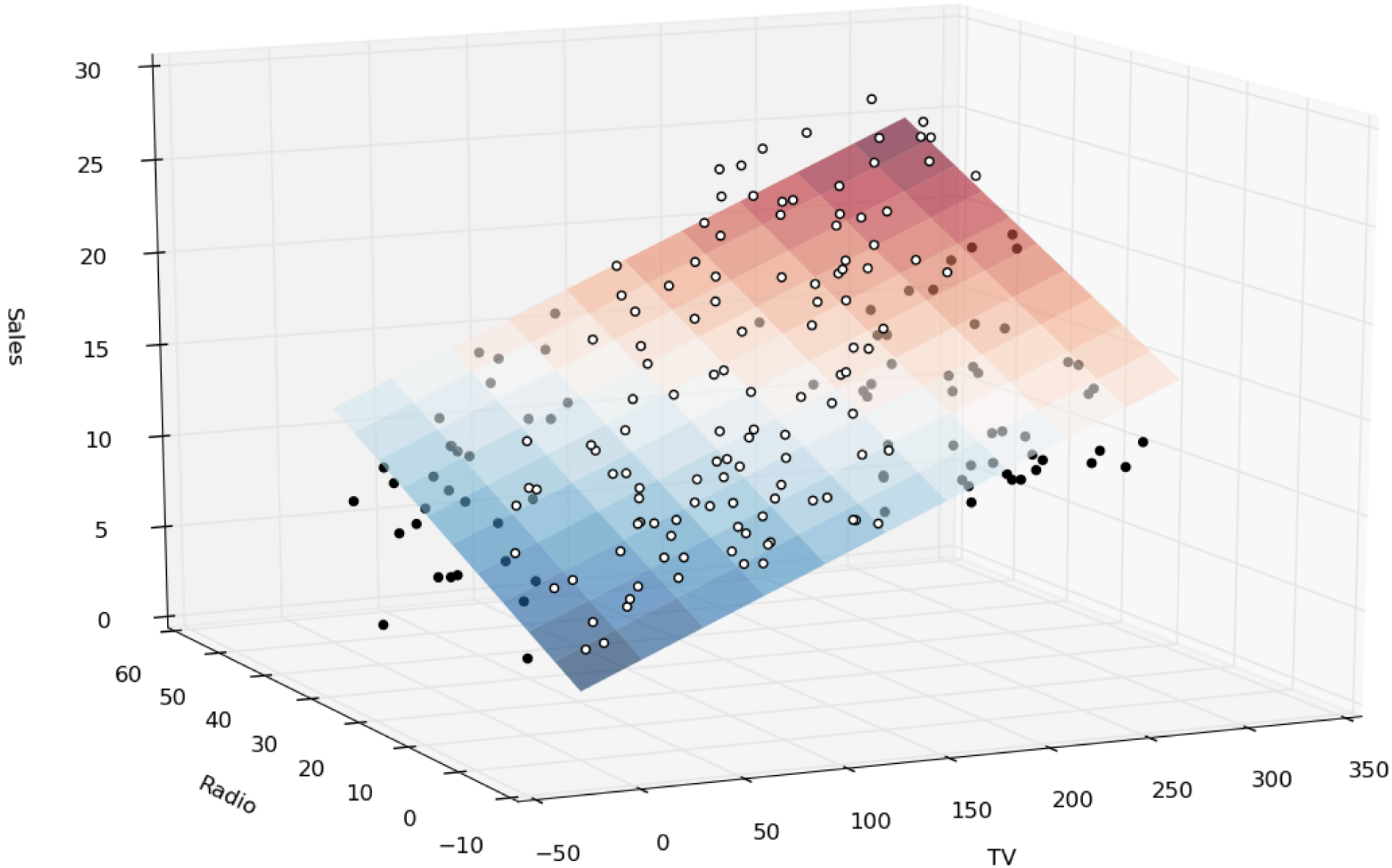
Statistical Regression With Python

Explain & Predict


15
10
5
-20
-10
10
20
30
40
50
60



Sales
30
25
20
15
10
5
0
Radio
60
50
40
30
20
10
0
−10
TV
−50
0
50
100
150
200
250
300
350

Explain & Predict

- $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$
- Explain by β, the slope.
- Predict by new x_i.
- "Simple linear regression model"

- $y_i = \mathbf{x}_i^\top \boldsymbol{\beta} + \varepsilon_i$
- $\boldsymbol{\beta}$, a slope vector.
- New $\boldsymbol{x}_i$, a vector.
- "Multiple linear regression model"

Mosky

- Python Charmer at Pinkoi.
- Has spoken at: PyCons in TW, MY, KR, JP, SG, HK, COSCUPs, and TEDx, etc.
- Countless hours on teaching Python.
- Own the Python packages: ZIPCodeTW, MoSQL, Clime, etc.
- http://mosky.tw/

Outline

- The Analysis Steps
  - Define Assumptions
  - Validate Assumptions
    - The Dataset: Fair
- Correlation Analysis
- Ordinary Least Squares
  - Models & Estimations
  - Understand Regression Result
  - Model Specification Using the R Formula
  - Covariance Types
  - Outliers
- Correlation & Causation
- More Models & Estimations
  - Introduction
  - Logit Model

The PDF, Notebooks, and Packages

- The PDF and notebooks are available on https://github.com/moskytw/statistical-regression-with-python .
- The packages:
  - `$ pip3 install jupyter numpy scipy sympy matplotlib ipython pandas seaborn statsmodels scikit-learn`

    Or:
  - `> conda install jupyter numpy scipy sympy matplotlib ipython pandas seaborn statsmodels scikit-learn`

Define Assumptions

- The regression analysis:
  - Suitable to measure the relationship between variables.
  - Can model most of the hypothesis testing. [ref]
  - Can predict, but machine learning methods may do better.

- “Years of marriage has association with children?”
- “Rates of marriage has association with affairs?”
- “Any background may have association with affairs?”

Validate Assumptions

- Collect data …
  - The “Fair” dataset:
    - Fair, Ray. 1978. “A Theory of Extramarital Affairs,” Journal of Political Economy, February, 45-61.
    - A dataset from 1970s.
    - Rows: 6,366
    - Columns: (next slide)
- The full version of the analysis steps: http://bit.ly/analysis-steps .

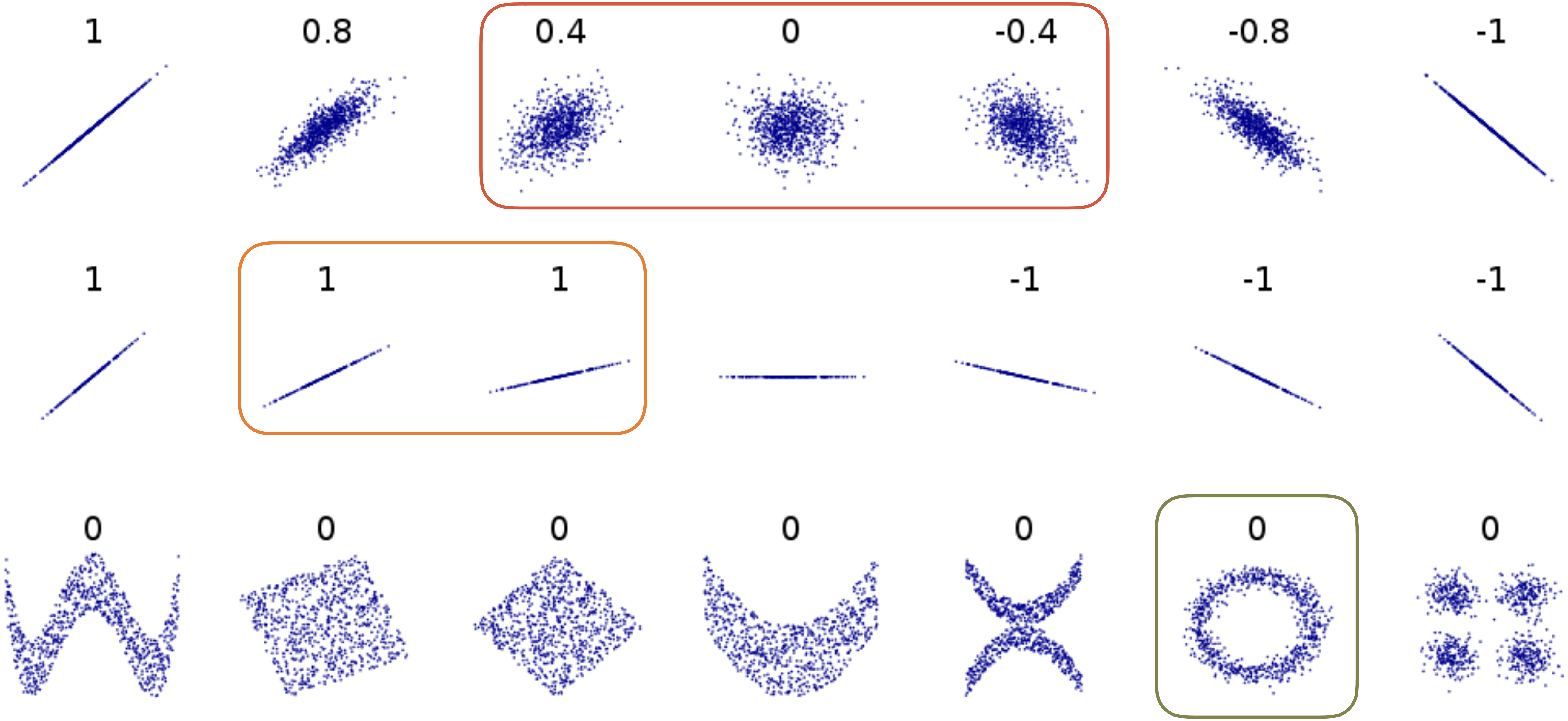
1. *rate_marriage*: 1~5; very poor, poor, fair, good, very good.
2. *age*
3. *yrs_married*
4. *children*: number of children.
5. *religious*: 1~4; not, mildly, fairly, strongly.
6. *educ*: 9, 12, 14, 16, 17, 20; grade school, some college, college graduate, some graduate school, advanced degree.
7. *occupation*: 1, 2, 3, 4, 5, 6; student, farming-like, white-colloar, teacher-like, business-like, professional with advanced degree.
8. *occupation_husb*
9. *affairs*: n times of extramarital affairs per year since marriage.

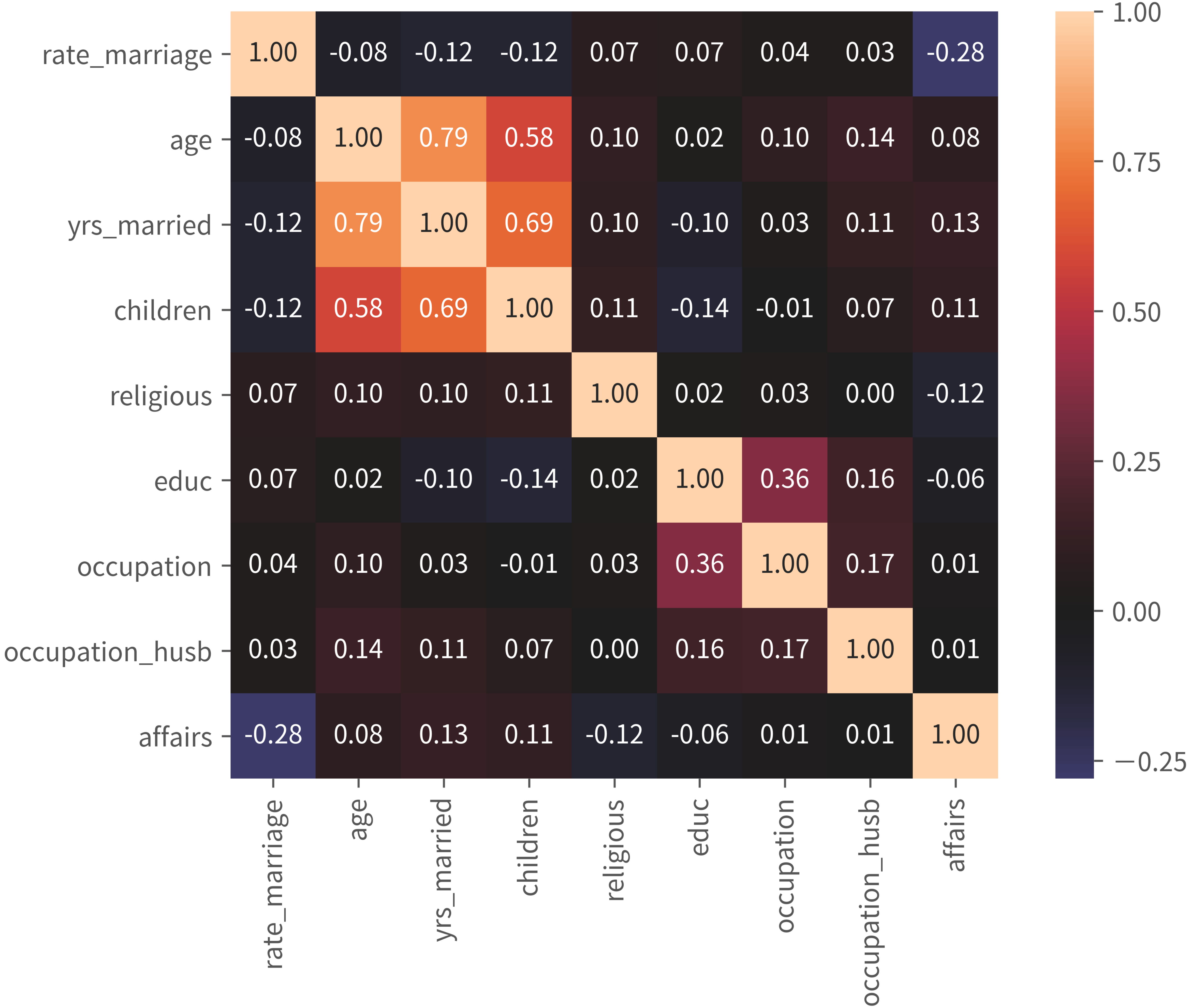

JESUS SAV
Fill up with the Holy Ghost & Fire

Correlation Analysis

- **Pearson correlation coefficient**
  - For the variables whose distance is meaningful.
  - `df.corr()`
- **Kendall rank correlation coefficient**
  - For the variables whose order is meaningful.
  - `df.corr('kendall')`
- Measures “tightness”.

Pearson Correlation Coefficient




rate_marriage
age
yrs_married
children
religious
educ
occupation
occupation_husb
affairs
1.00 -0.08 -0.12 -0.12 0.07 0.07 0.04 0.03 -0.28
-0.08 1.00 0.79 0.58 0.10 0.02 0.10 0.14 0.08
-0.12 0.79 1.00 0.69 0.10 -0.10 0.03 0.11 0.13
-0.12 0.58 0.69 1.00 0.11 -0.14 -0.01 0.07 0.11
0.07 0.10 0.10 0.11 1.00 0.02 0.03 0.00 -0.12
0.07 0.02 -0.10 -0.14 0.02 1.00 0.36 0.16 -0.06
0.04 0.10 0.03 -0.01 0.03 0.36 1.00 0.17 0.01
0.03 0.14 0.11 0.07 0.00 0.16 0.17 1.00 0.01
-0.28 0.08 0.13 0.11 -0.12 -0.06 0.01 0.01 1.00
1.00
0.75
0.50
0.25
0.00
−0.25

```python
import statsmodels.api as sm
import seaborn as sns


print(sm.datasets.fair.SOURCE,
      sm.datasets.fair.NOTE)

# -> Pandas's Dataframe
df_fair = sm.datasets.fair.load_pandas().data


df = df_fair
sns.heatmap(df.corr(method='kendall'),
            center=0, square=True,
            annot=True, fmt='.2f')
```

Models & Estimations

- **Models**
  - $y = X\beta + \varepsilon$
  - Like simple, multiple, logit, etc.
- **Estimations**
  - How to estimate the $\hat{\beta}$? For example, OLS:

$$y = X\hat{\beta}$$

$$S(b) = \sum_{i=1}^{n} (y_i - x_i^{\mathrm{T}} b)^2 = (y - Xb)^{\mathrm{T}}(y - Xb)$$

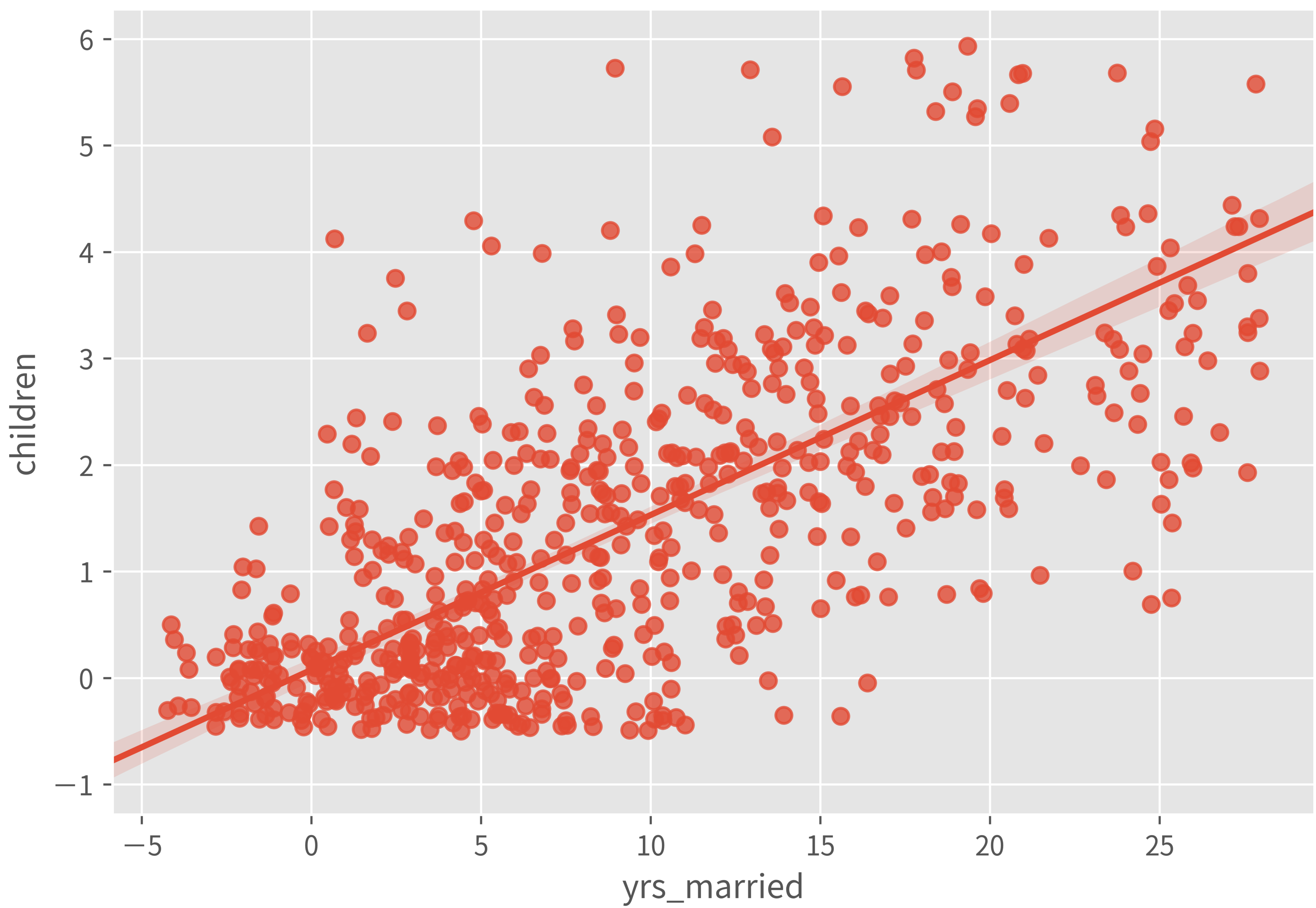
$$\hat{\beta} = \operatorname{argmin}_{b \in \mathbb{R}^p} S(b) = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$$

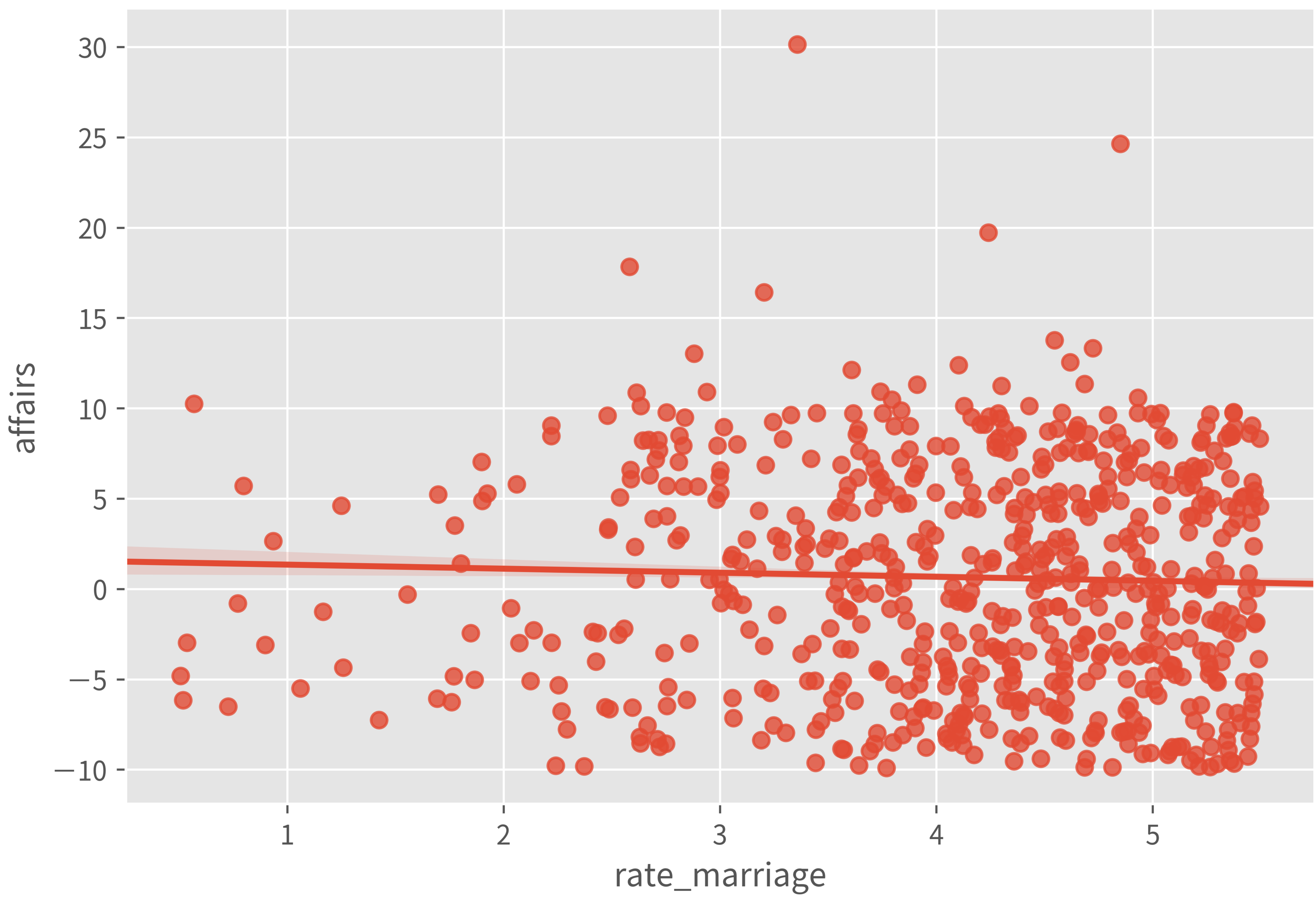
Model Specification Using the R Formula

- Using the R formula implementation in Python, Patsy:

$$
\begin{aligned}
& y \sim x \\
\equiv\ & y \sim 1 + x \\
\equiv\ & y = \beta_0 1 + \beta_1 x + \varepsilon
\end{aligned}
$$

- For example:
  - `affairs ~ rate_marriage`


children
yrs_married
−1
0
1
2
3
4
5
6
−5
0
5
10
15
20
25



30
25
20
15
10
5
0
−5
−10
affairs
1
2
3
4
5
rate_marriage

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

```
df_fair_sample = df_fair.sample(
    frac=0.1, random_state=20190425
)
```

```
df = df_fair_sample
sns.regplot(data=df, x='yrs_married', y='children',
            x_jitter=10/2, y_jitter=1/2)
```

```
df = df_fair_sample
sns.regplot(data=df, x='rate_marriage', y='affairs',
            x_jitter=1/2, y_jitter=20/2)
```

```
df = df_fair
(smf
 .ols('affairs ~ rate_marriage', df)
 .fit()
 .summary())
```

Adj. R-squared

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

- ≡ *explained var. by X / var. of y and adjusted by no. of X*
- $\in [0, 1]$, usually.
- Can compare among models.
- 0.032 is super bad.

Prob(F-statistics)

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

- ≡ $P(\mathit{data} \mid \mathit{all\ coefs\ are\ zero})$
- Trust the coefs if low enough.
- "Low enough" is "< 0.05" in convention.

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

Log-Likelihood

- Higher is better.
- Negative, usually.
- Can compare among models when the datasets are the same.
- Also check likelihood-ratio test.

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

Large Sample or Normality

- To construct interval estimates correctly, e.g., hypothesis tests on coefs, confidence intervals.
- **No. Observations**
  - $\geq$ 110~200 [ref]
- **Normality of Residuals**
  - Prob($Omnibus$) ≥ 0.05
  - $\wedge$ Prob(JB) ≥ 0.05

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

Cond. No.

- Measures the degree of multicollinearity.
- Multicollinearity increases the std err, i.e., decreases efficiency.
- If ≥ 30, check:
  - Any variable has redundant information? Like fat % and weight. Drop one.
  - Dummy variable trap?
  - If no, the model is good.
  - Other suggestions.

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

P>|t|

- $\equiv P(\textit{data} \mid \textit{the coef is zero})$
- Exclude the *x* whose p-value is not low enough.

Coef & Confidence Intervals

- "The rate_marriage and affairs has negative relationship, the strength is -0.4081, and 95% confidence interval is [-0.464, -0.353]."

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

Code Categorical Variables

- The str or bool is treated as categorical by default.
- Or use the C function:

$$
\begin{aligned}
& y \sim \mathrm{C}(c) \\
\equiv\ & y \sim 1 + (\cancel{c_1} + c_2 + \ldots + c_i) - \cancel{c_1} \\
\equiv\ & y = \beta_0 1 + (\cancel{\beta_1 c_1} + \beta_2 c_2 + \ldots + \beta_i c_i) - \cancel{\beta_1 c_1} + \varepsilon
\end{aligned}
$$

- The x_1 is chosen as reference level automatically.
- For example:

$$
\begin{aligned}
& \mathrm{C}(\mathit{rate\_marriage} \in \{1, 2, 3, 4, 5\}) \\
\equiv\ & 1 + \mathit{rate\_marriage\_2} \in \{0, 1\} + \ldots + \mathit{rate\_marriage\_5} \in \{0, 1\}
\end{aligned}
$$

affairs ~ C(rate_marriage)

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.036
Model:	OLS	**Adj. R-squared:**	0.035
Method:	Least Squares	**F-statistic:**	58.57
Date:	Sat, 27 Apr 2019	**Prob (F-statistic):**	1.25e-48
Time:	15:27:20	**Log-Likelihood:**	-13946.
No. Observations:	6366	**AIC:**	2.790e+04
Df Residuals:	6361	**BIC:**	2.794e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	1.2017	0.218	5.524	0.000	0.775	1.628
C(rate_marriage)[T.2.0]	0.4141	0.247	1.679	0.093	-0.069	0.897
C(rate_marriage)[T.3.0]	0.1696	0.228	0.743	0.457	-0.278	0.617
C(rate_marriage)[T.4.0]	-0.5268	0.222	-2.370	0.018	-0.963	-0.091
C(rate_marriage)[T.5.0]	-0.8535	0.222	-3.853	0.000	-1.288	-0.419

Omnibus:	9436.269	**Durbin-Watson:**	1.612
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5218576.884
Skew:	8.915	**Prob(JB):**	0.00
Kurtosis:	142.127	**Cond. No.**	21.0

- If 1, affairs is 1.2017.
- If 5, affairs is 1.2017-0.8535.
- The 2, 3, 4 are not significant to the reference level.

affairs ~ 0 + C(rate_marriage)

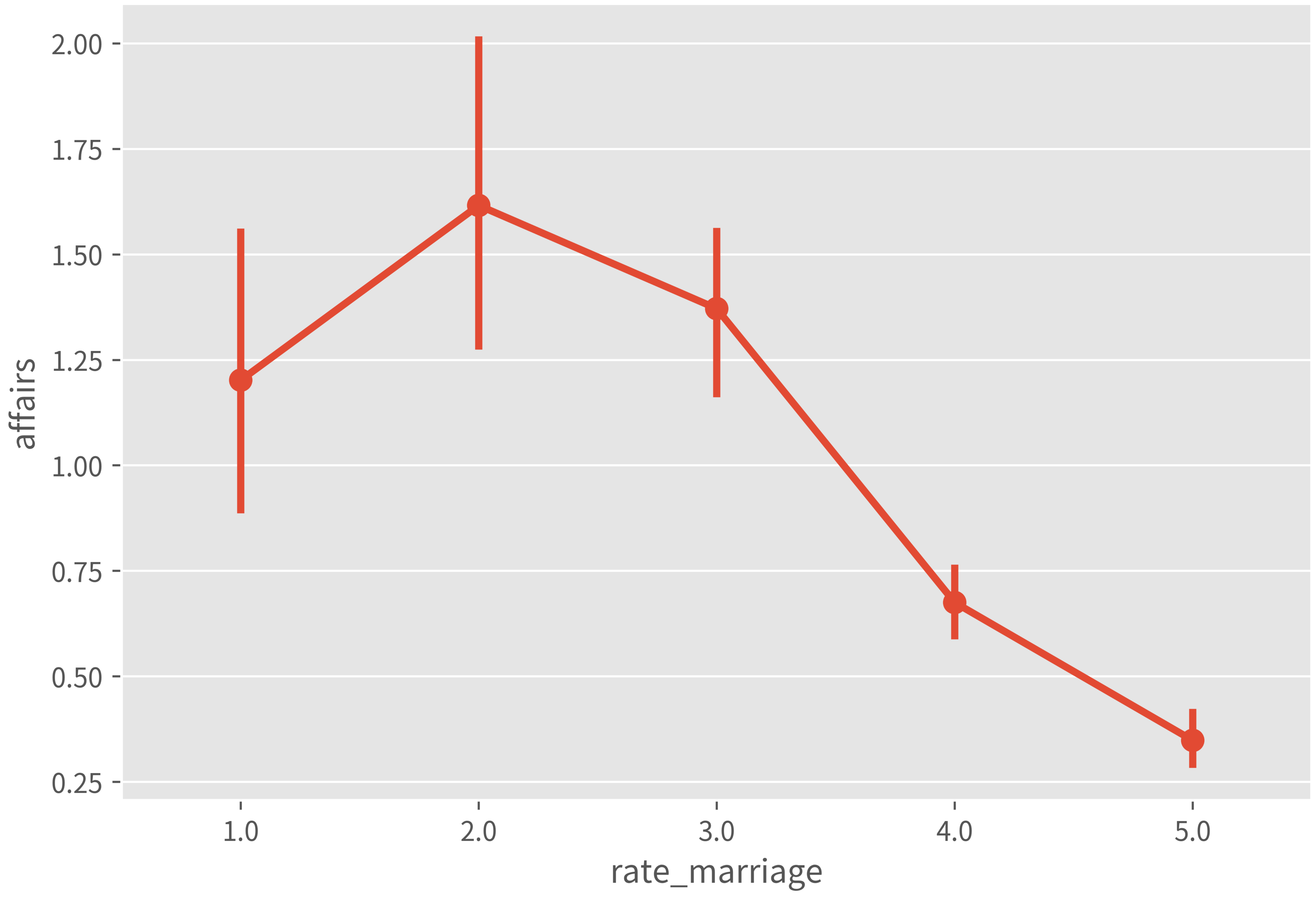
OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.036
Model:	OLS	**Adj. R-squared:**	0.035
Method:	Least Squares	**F-statistic:**	58.57
Date:	Sat, 27 Apr 2019	**Prob (F-statistic):**	1.25e-48
Time:	17:20:18	**Log-Likelihood:**	-13946.
No. Observations:	6366	**AIC:**	2.790e+04
Df Residuals:	6361	**BIC:**	2.794e+04
Df Model:	4		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
C(rate_marriage)[1.0]	1.2017	0.218	5.524	0.000	0.775	1.628
C(rate_marriage)[2.0]	1.6157	0.116	13.925	0.000	1.388	1.843
C(rate_marriage)[3.0]	1.3713	0.069	19.963	0.000	1.237	1.506
C(rate_marriage)[4.0]	0.6748	0.046	14.762	0.000	0.585	0.764
C(rate_marriage)[5.0]	0.3482	0.042	8.333	0.000	0.266	0.430

Omnibus:	9436.269	**Durbin-Watson:**	1.612
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5218576.884
Skew:	8.915	**Prob(JB):**	0.00
Kurtosis:	142.127	**Cond. No.**	5.21

- `y ~ 0 + c`
  - Code without a reference.
- To calculate the mean of each group.



2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
affairs
1.0
2.0
3.0
4.0
5.0
rate_marriage

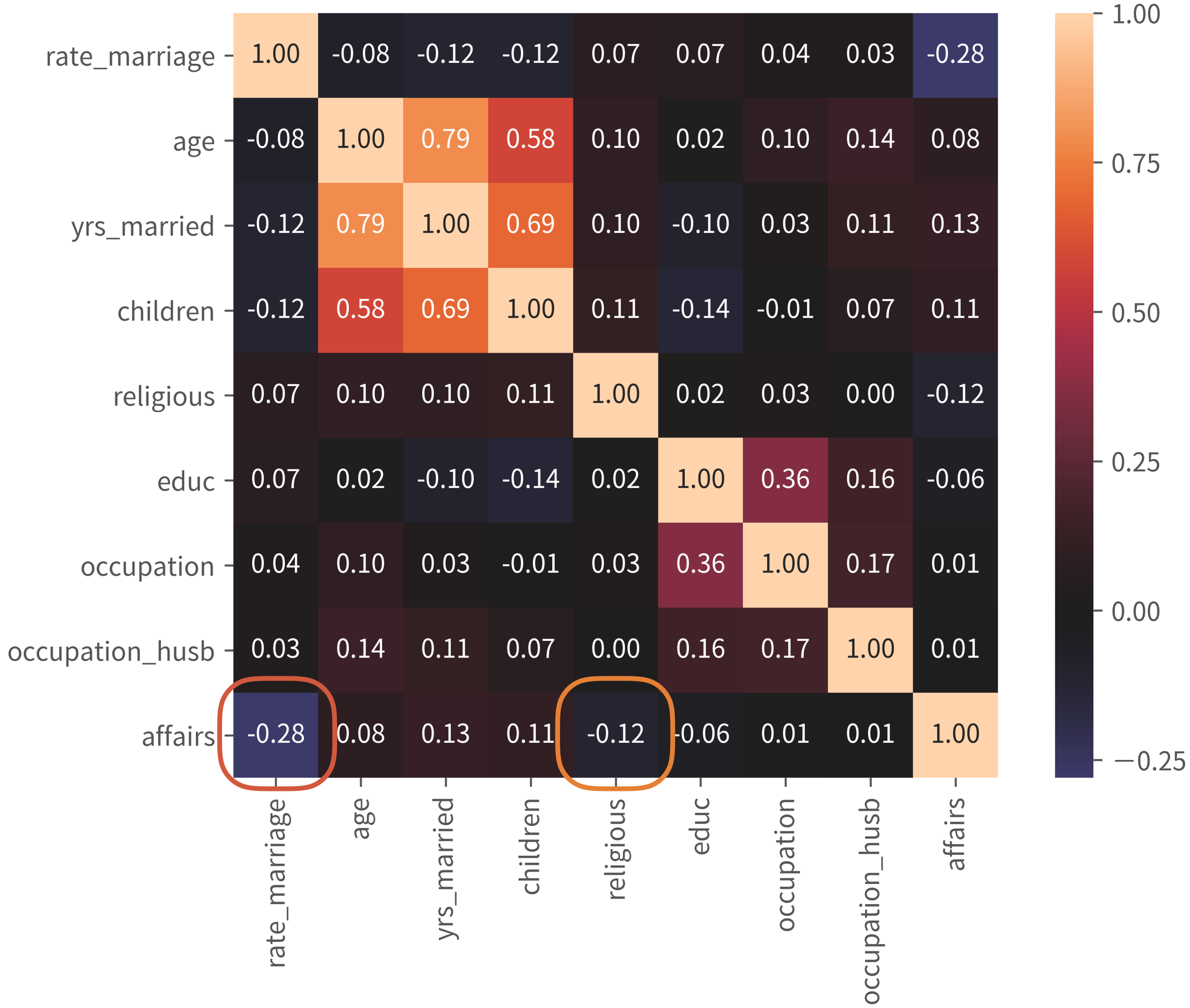
```python
df = df_fair
sns.pointplot(data=df, x='rate_marriage', y='affairs')
```

```python
df = df_fair
(smf
 .ols('affairs ~ C(rate_marriage)', df)
 .fit()
 .summary())
```

```python
df = df_fair
(smf
 .ols('affairs ~ 0 + C(rate_marriage)', df)
 .fit()
 .summary())
```

Other Ways to Code Categorical Variables

- `y ~ C(x, Treatment(reference='A'))`
  - Specify the reference level.
- `affairs ~ 0 + C(rate_marriage, Diff)`
  - Compare each level with the preceding level.
- `affairs ~ 0 + C(rate_marriage, Sum)`
  - Compare each level with the mean-of-means.
- Check the full reference.



Interaction

- "The low rate_marriage with high religious has **stronger** negative relationship with affairs?"

$$y \sim x * z$$

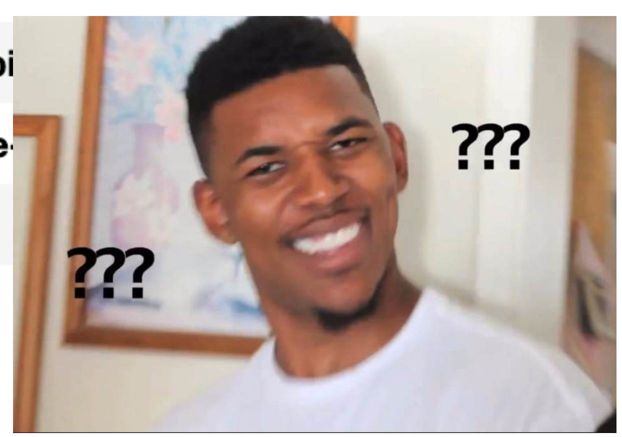
$$\equiv y = \beta_0 1 + \beta_1 x + \beta_2 z + \beta_3 xz + \varepsilon$$

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.048
Model:	OLS	**Adj. R-squared:**	0.048
Method:	Least Squares	**F-statistic:**	107.8
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	4.55e-68
Time:	23:25:03	**Log-Likelihood:**	-13904.
No. Observations:	6366	**AIC:**	2.782e+04
Df Residuals:	6362	**BIC:**	2.784e+04
Df Model:	3		
Covariance Type:	nonrobust		

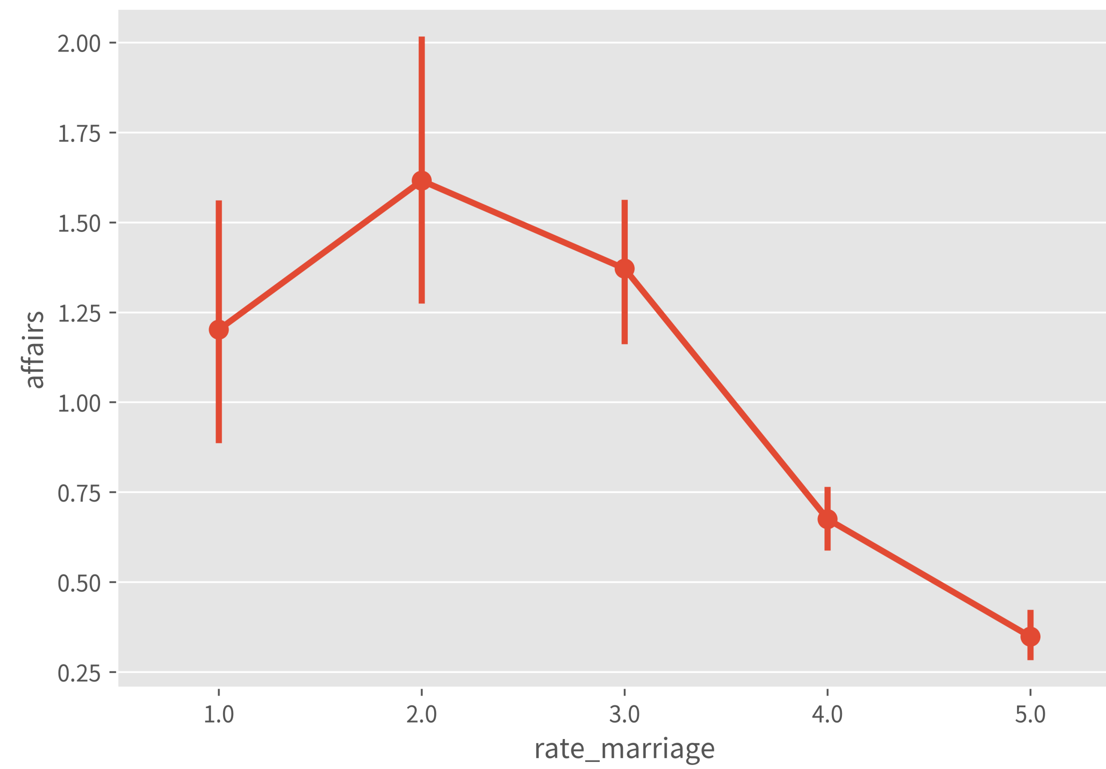
	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	4.6495	0.347	13.395	0.000	3.969	5.330
rate_marriage	-0.7891	0.082	-9.622	0.000	-0.950	-0.628
religious	-0.9846	0.138	-7.122	0.000	-1.256	-0.714
rate_marriage:religious	0.1681	0.032	5.209	0.000	0.105	0.231

Omnibus:	9399.882	**Durbi**	
Prob(Omnibus):	0.000	**Jarque-**	
Skew:	8.843		
Kurtosis:	141.848		



affairs ~ rate_marriage*religious

- The model may be wrong, since the relationship is not linear.



	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	1.3341	0.498	2.678	0.007	0.358	2.311
C(rate_marriage)[T.2.0]	1.9654	0.921	2.134	0.033	0.160	3.771
C(rate_marriage)[T.3.0]	1.0479	0.634	1.654	0.098	-0.194	2.290
C(rate_marriage)[T.4.0]	-0.3282	0.517	-0.635	0.525	-1.341	0.685
C(rate_marriage)[T.5.0]	-0.6430	0.522	-1.232	0.218	-1.666	0.380
C(religious)[T.2.0]	0.1143	0.627	0.182	0.855	-1.115	1.343
C(religious)[T.3.0]	-0.3413	0.529	-0.646	0.519	-1.377	0.695
C(religious)[T.4.0]	-0.6082	0.588	-1.035	0.301	-1.760	0.544
C(rate_marriage)[T.2.0]:C(religious)[T.2.0]	-1.8103	1.028	-1.761	0.078	-3.825	0.204
C(rate_marriage)[T.3.0]:C(religious)[T.2.0]	-1.1905	0.754	-1.580	0.114	-2.668	0.286
C(rate_marriage)[T.4.0]:C(religious)[T.2.0]	-0.3499	0.646	-0.542	0.588	-1.615	0.916
C(rate_marriage)[T.5.0]:C(religious)[T.2.0]	-0.4682	0.647	-0.723	0.469	-1.737	0.800
C(rate_marriage)[T.2.0]:C(religious)[T.3.0]	-1.8707	0.950	-1.968	0.049	-3.734	-0.008
C(rate_marriage)[T.3.0]:C(religious)[T.3.0]	-0.9741	0.665	-1.464	0.143	-2.278	0.330
C(rate_marriage)[T.4.0]:C(religious)[T.3.0]	-0.1333	0.549	-0.243	0.808	-1.209	0.943
C(rate_marriage)[T.5.0]:C(religious)[T.3.0]	-0.0448	0.553	-0.081	0.935	-1.128	1.039
C(rate_marriage)[T.2.0]:C(religious)[T.4.0]	-2.2192	0.997	-2.225	0.026	-4.174	-0.265
C(rate_marriage)[T.3.0]:C(religious)[T.4.0]	-1.1008	0.726	-1.517	0.129	-2.523	0.321
C(rate_marriage)[T.4.0]:C(religious)[T.4.0]	-0.0946	0.608	-0.156	0.876	-1.286	1.097
C(rate_marriage)[T.5.0]:C(religious)[T.4.0]	0.0196	0.608	0.032	0.974	-1.173	1.212

affairs ~ C(rate_marriage)*C(religious)

- Hmmm ...
- TL;DR by ANOVA.

	sum_sq	df	F	PR(>F)
Intercept	32.952681	1.0	7.171451	0.007426
C(rate_marriage)	117.831212	4.0	6.410865	0.000038
C(religious)	11.838124	3.0	0.858772	0.461713
C(rate_marriage):C(religious)	111.850861	12.0	2.028497	0.018427
Residual	29159.748241	6346.0	NaN	NaN

- Looks like C(*religious*) isn't helping to explain in this model. Drop it.

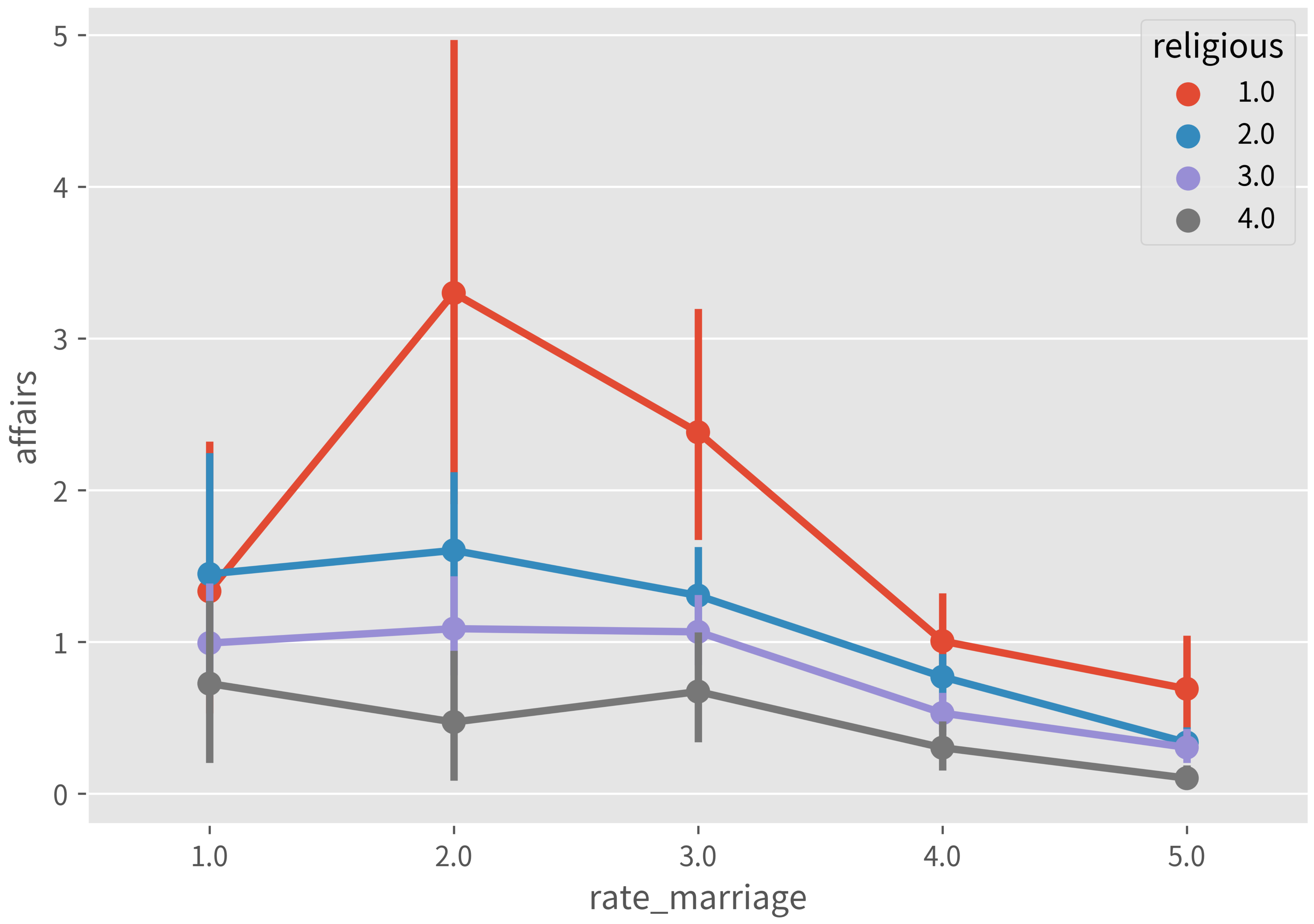
```python
res = res_1
df = pd.DataFrame(dict(params=res.params,
                       pvalues=res.pvalues))
df[df.pvalues < 0.05].sort_values('params')
```

	params	pvalues
C(rate_marriage)[2.0]:C(religious)[T.4.0]	-2.827311	0.000450
C(rate_marriage)[2.0]:C(religious)[T.3.0]	-2.211959	0.005104
C(rate_marriage)[3.0]:C(religious)[T.4.0]	-1.708937	0.000059
C(rate_marriage)[2.0]:C(religious)[T.2.0]	-1.695969	0.037348
C(rate_marriage)[3.0]:C(religious)[T.3.0]	-1.315414	0.001136
C(rate_marriage)[3.0]:C(religious)[T.2.0]	-1.076195	0.010052
C(rate_marriage)[4.0]:C(religious)[T.4.0]	-0.702780	0.000006
C(rate_marriage)[5.0]:C(religious)[T.4.0]	-0.588575	0.000194
C(rate_marriage)[4.0]:C(religious)[T.3.0]	-0.474564	0.001321
C(rate_marriage)[5.0]:C(religious)[T.3.0]	-0.386073	0.016820
C(rate_marriage)[5.0]:C(religious)[T.2.0]	-0.353891	0.027721
Intercept	1.334104	0.007407
C(rate_marriage)[T.2.0]	1.965381	0.032863

affairs ~ C(rate_marriage) + C(R):C(L)

$$y \sim x : z$$
$$\equiv y = \beta_0 + \beta_1 xz + \varepsilon$$

- Stronger!
- ≡ `A ~ C(R)*C(L) - C(L)`


religious
1.0
2.0
3.0
4.0
affairs
0
1
2
3
4
5
1.0
2.0
3.0
4.0
5.0
rate_marriage

```
df = df_fair
(smf
 .ols('affairs ~ rate_marriage*religious', df)
 .fit()
 .summary())
```

```
df = df_fair
res = (smf
       .ols('affairs'
            '~ C(rate_marriage)*C(religious)', df)
       .fit())
display(res.summary(),
        # type III is suitable to unbalanced dataset
        # ref: http://bit.ly/3typess
        sm.stats.anova_lm(res, typ=3))
```

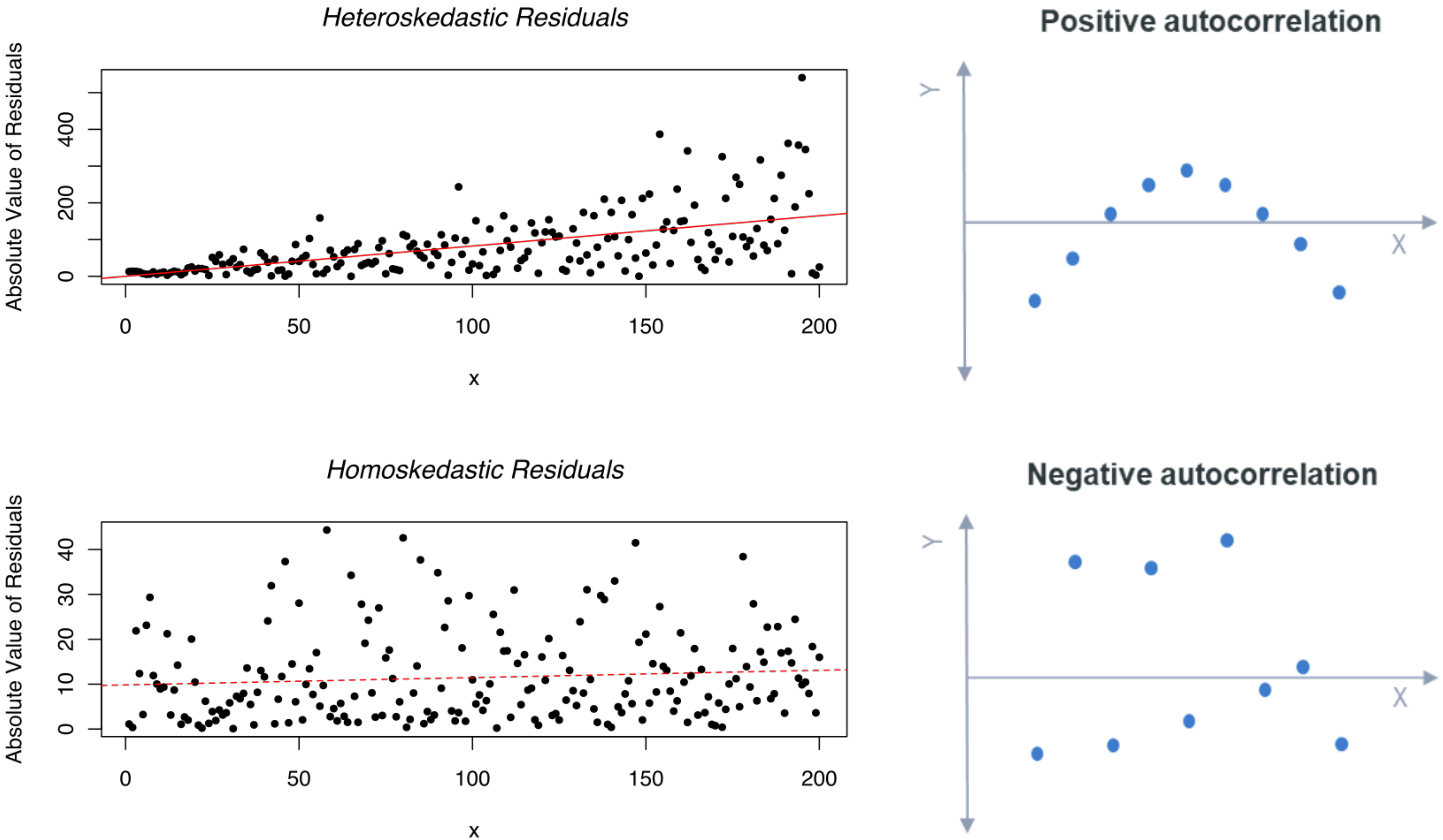
```python
df = df_fair
res = (smf
       .ols('affairs'
            '~ C(rate_marriage)'
            '+ C(rate_marriage):C(religious)', df)
       .fit())
display(res.summary(),
        sm.stats.anova_lm(res, typ=3))
```

```python
df = df_fair
sns.pointplot(data=df,
              x='rate_marriage',
              y='affairs',
              hue='religious')
```

More Operators: Transformation & Control Variables

- `np.log(y) ~ x`
  - If y and x have a better linear relationship after transform.
  - Note that:
    - $\log(y) = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2$
    - $y = \exp(\hat{\beta}_1 x_1 + \hat{\beta}_2 x_2)$
    - $y = \exp(\hat{\beta}_1 x_1) \times \exp(\hat{\beta}_2 x_2)$
- `np.sqrt(y) ~ x`
- `y ~ I(x*z)`
  - True multiplication.

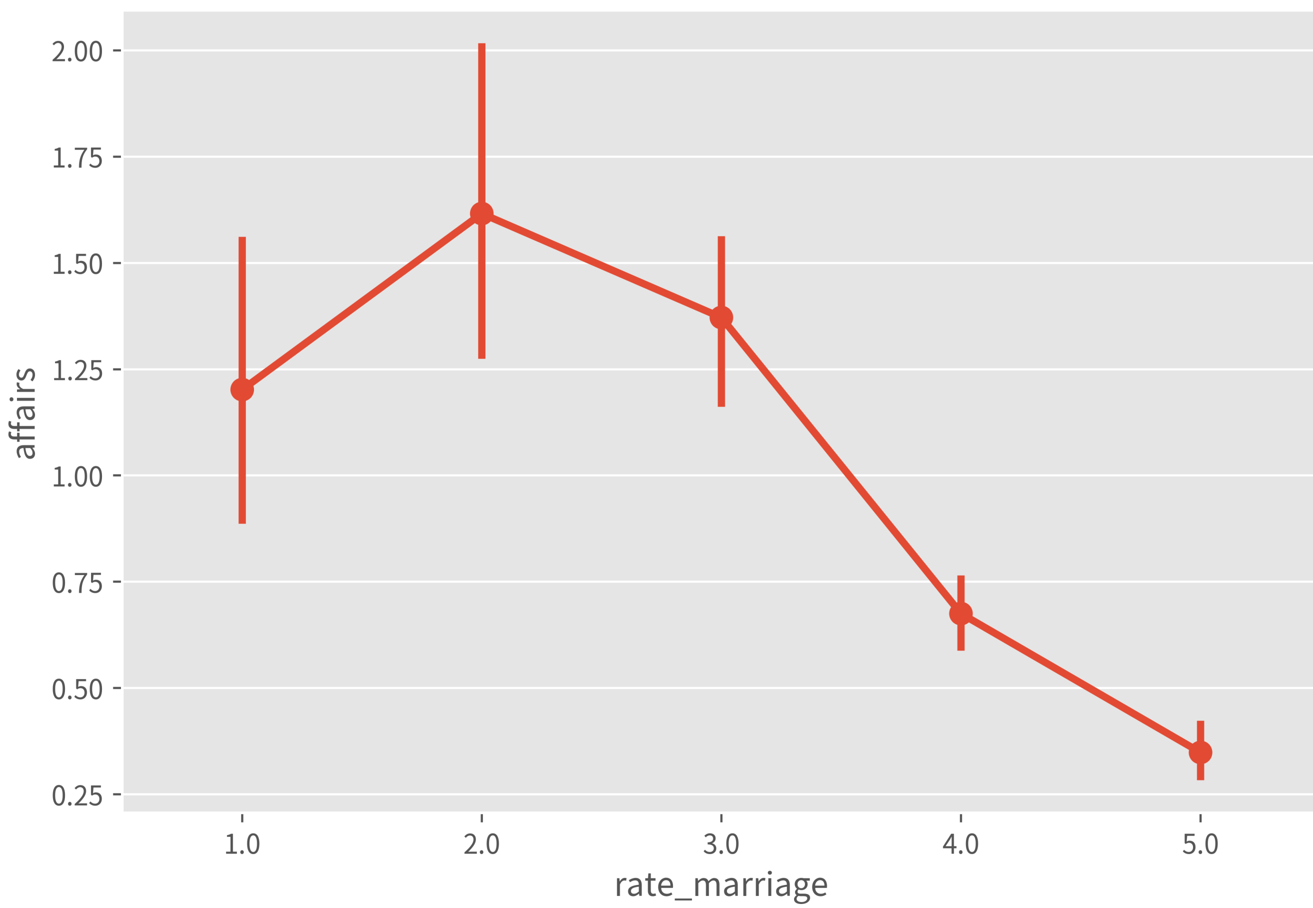
- `y ~ z_1 + ... + x_1 + ...`
  - z_i and x_i are both independent variables.
  - If we don't interest in z_i, but add them to carry some effects and clarify the effects of x_i, we call z_i "control variables".
  - For example:
    - `conversion_rate ~ month + group`
- Check the full reference.



Covariance Types of Errors

Covariance Types of Errors

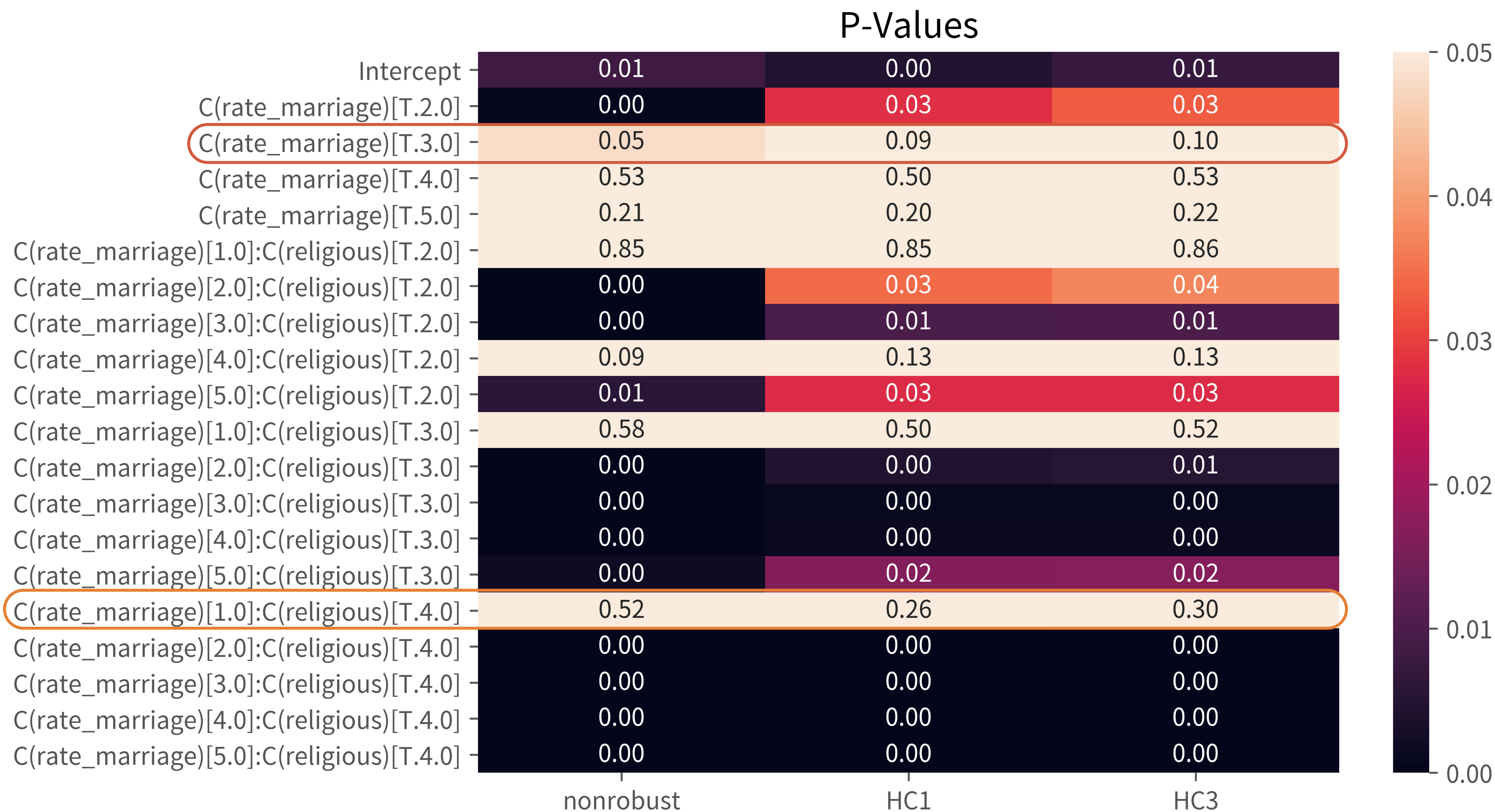
- Spherical Errors
  - ≡ Homoscedasticity & no autocorrelation
- Heteroscedasticity
- Autocorrelation ≡ serial correlation
- If spherical errors, the model is good.
- If **not** spherical errors, the std errs are wrong.
  - So the interval estimates are **wrong**, including **hypothesis tests** on coefs, **confidence intervals**.



Heteroscedasticity

- Use HC std errs (heteroscedasticity-consistent standard errors) to correct.
- If $N \leq 250$, use HC3. [ref]
- If $N > 250$, consider HC1 for the speed.
- Also suggest to use by default.
- `.fit(cov_type='HC3')`

← The confidence intervals vary among groups. The heteroscedasticity exists.


P-Values
Intercept
C(rate_marriage)[T.2.0]
C(rate_marriage)[T.3.0]
C(rate_marriage)[T.4.0]
C(rate_marriage)[T.5.0]
C(rate_marriage)[1.0]:C(religious)[T.2.0]
C(rate_marriage)[2.0]:C(religious)[T.2.0]
C(rate_marriage)[3.0]:C(religious)[T.2.0]
C(rate_marriage)[4.0]:C(religious)[T.2.0]
C(rate_marriage)[5.0]:C(religious)[T.2.0]
C(rate_marriage)[1.0]:C(religious)[T.3.0]
C(rate_marriage)[2.0]:C(religious)[T.3.0]
C(rate_marriage)[3.0]:C(religious)[T.3.0]
C(rate_marriage)[4.0]:C(religious)[T.3.0]
C(rate_marriage)[5.0]:C(religious)[T.3.0]
C(rate_marriage)[1.0]:C(religious)[T.4.0]
C(rate_marriage)[2.0]:C(religious)[T.4.0]
C(rate_marriage)[3.0]:C(religious)[T.4.0]
C(rate_marriage)[4.0]:C(religious)[T.4.0]
C(rate_marriage)[5.0]:C(religious)[T.4.0]
nonrobust
HC1
HC3
0.01 0.00 0.01
0.00 0.03 0.03
0.05 0.09 0.10
0.53 0.50 0.53
0.21 0.20 0.22
0.85 0.85 0.86
0.00 0.03 0.04
0.00 0.01 0.01
0.09 0.13 0.13
0.01 0.03 0.03
0.58 0.50 0.52
0.00 0.00 0.01
0.00 0.00 0.00
0.00 0.00 0.00
0.00 0.02 0.02
0.52 0.26 0.30
0.00 0.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00
0.00 0.00 0.00
0.05
0.04
0.03
0.02
0.01
0.00

OLS Regression Results

Dep. Variable:	affairs	**R-squared:**	0.032
Model:	OLS	**Adj. R-squared:**	0.032
Method:	Least Squares	**F-statistic:**	208.4
Date:	Fri, 26 Apr 2019	**Prob (F-statistic):**	1.66e-46
Time:	23:25:02	**Log-Likelihood:**	-13959.
No. Observations:	6366	**AIC:**	2.792e+04
Df Residuals:	6364	**BIC:**	2.794e+04
Df Model:	1		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	2.3825	0.119	19.969	0.000	2.149	2.616
rate_marriage	-0.4081	0.028	-14.436	0.000	-0.464	-0.353

Omnibus:	9443.528	**Durbin-Watson:**	1.606
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	5215639.758
Skew:	8.930	**Prob(JB):**	0.00
Kurtosis:	142.083	**Cond. No.**	19.5

Autocorrelation

- Durbin-Watson
  - 2 is no autocorrelation.
  - [0, 2) is positive autocorrelation.
  - (2, 4] is negative autocorrelation.
  - [1.5, 2.5] are relatively normal. [ref]
- Use HAC std err.
- `.fit(cov_type='HAC', cov_kwds=dict(maxlag=tau))`

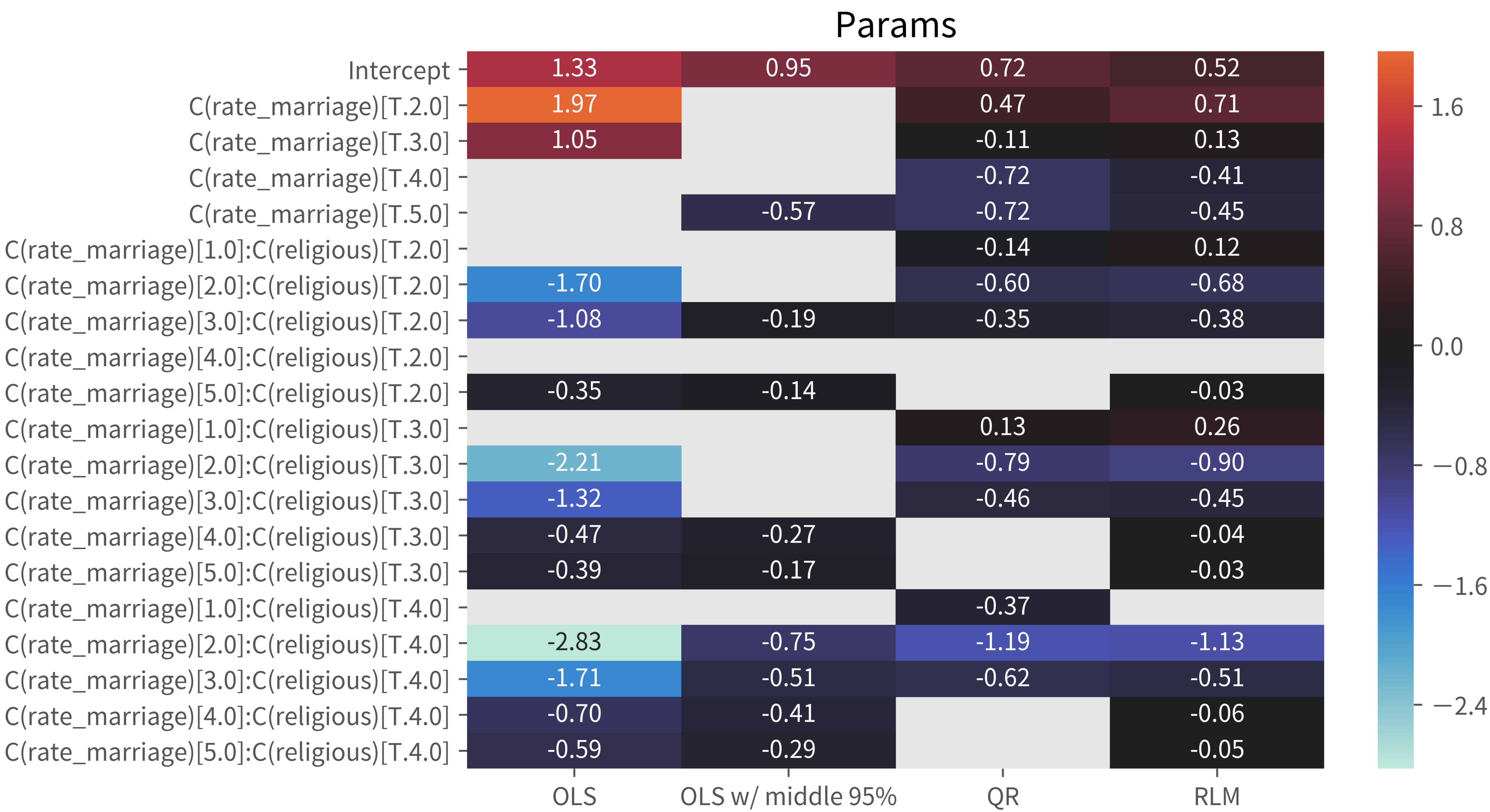
Other Covariance Types

- `cluster`
  - Assume each group has spherical errors.
- `hac-groupsum`
  - Sum by the time label and then process.
- `hac-panel`
  - Process by the groups and then aggregate.
- Check the full references.

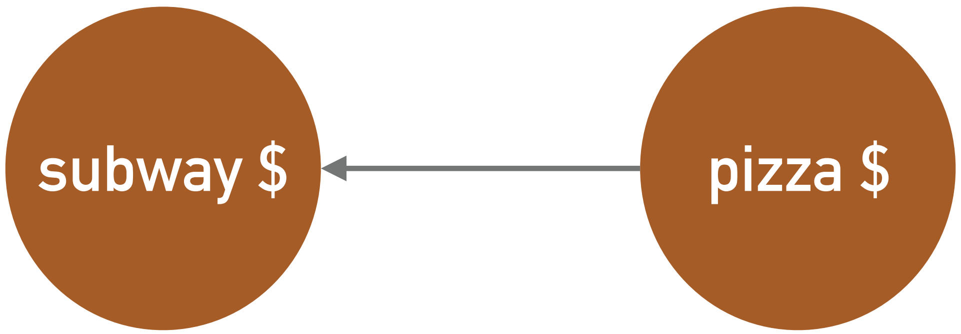
Outliers

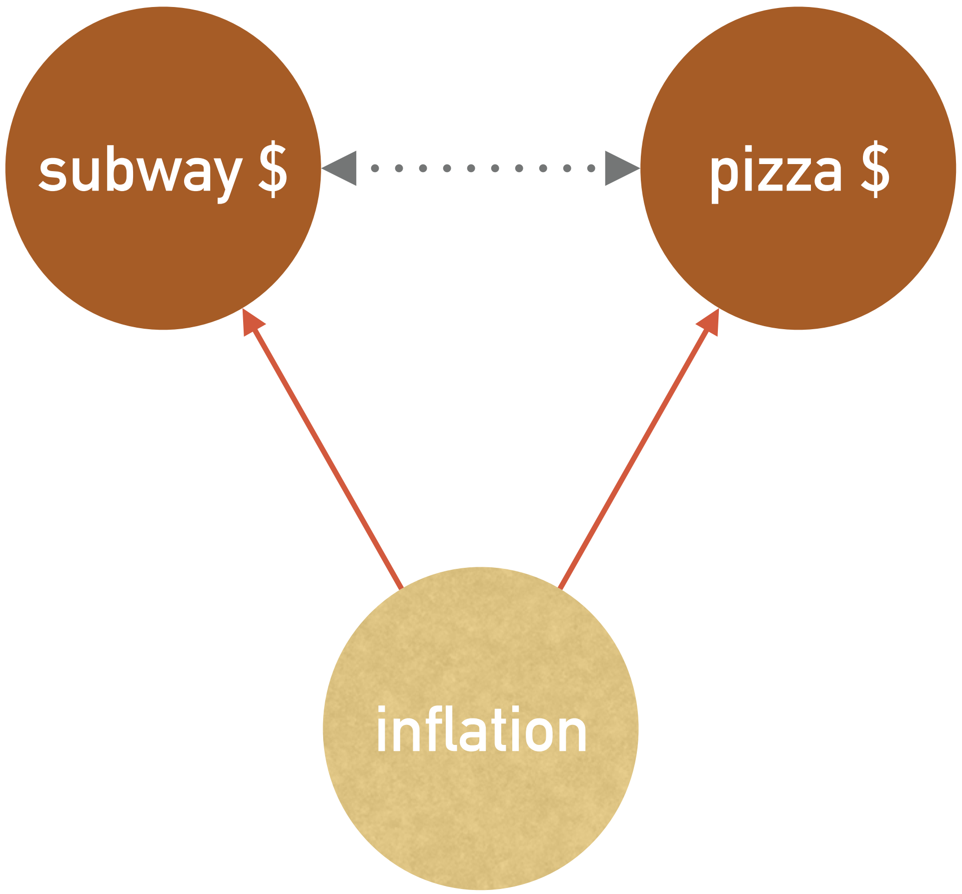
- An outlier may be the most interesting observation.
- Consider to include more variables to explain.
- Consider the possible non-linear relationship.
- Consider outlier-insensitive models.
- Do not drop observation without a good reason.
- Some good reasons:
  - Typo.
  - Not representative of the intended study population.
- Report the models with and without outliers.
- Include the preprocess steps in the report.

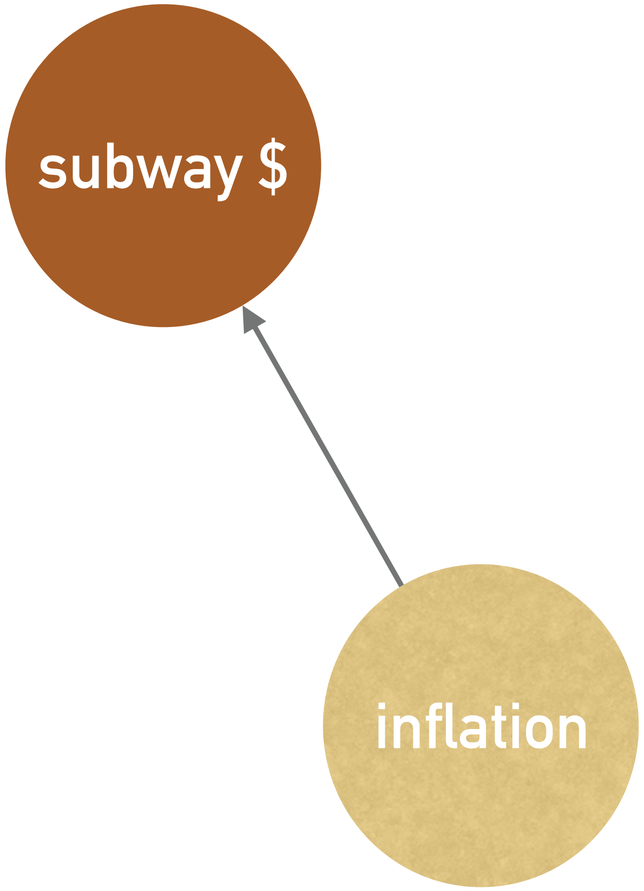
- Quantile regression: estimates the median rather than mean.
- Robust regression: robust to outliers, but slower.
- Keep middle n%: changes the intended study population.
- OLS
  - Outliner test
  - Influence report


Params
Intercept
C(rate_marriage)[T.2.0]
C(rate_marriage)[T.3.0]
C(rate_marriage)[T.4.0]
C(rate_marriage)[T.5.0]
C(rate_marriage)[1.0]:C(religious)[T.2.0]
C(rate_marriage)[2.0]:C(religious)[T.2.0]
C(rate_marriage)[3.0]:C(religious)[T.2.0]
C(rate_marriage)[4.0]:C(religious)[T.2.0]
C(rate_marriage)[5.0]:C(religious)[T.2.0]
C(rate_marriage)[1.0]:C(religious)[T.3.0]
C(rate_marriage)[2.0]:C(religious)[T.3.0]
C(rate_marriage)[3.0]:C(religious)[T.3.0]
C(rate_marriage)[4.0]:C(religious)[T.3.0]
C(rate_marriage)[5.0]:C(religious)[T.3.0]
C(rate_marriage)[1.0]:C(religious)[T.4.0]
C(rate_marriage)[2.0]:C(religious)[T.4.0]
C(rate_marriage)[3.0]:C(religious)[T.4.0]
C(rate_marriage)[4.0]:C(religious)[T.4.0]
C(rate_marriage)[5.0]:C(religious)[T.4.0]
OLS
OLS w/ middle 95%
QR
RLM
1.33 0.95 0.72 0.52
1.97 0.47 0.71
1.05 -0.11 0.13
-0.72 -0.41
-0.57 -0.72 -0.45
-0.14 0.12
-1.70 -0.60 -0.68
-1.08 -0.19 -0.35 -0.38
-0.35 -0.14 -0.03
0.13 0.26
-2.21 -0.79 -0.90
-1.32 -0.46 -0.45
-0.47 -0.27 -0.04
-0.39 -0.17 -0.03
-0.37
-2.83 -0.75 -1.19 -1.13
-1.71 -0.51 -0.62 -0.51
-0.70 -0.41 -0.06
-0.59 -0.29 -0.05
1.6
0.8
0.0
−0.8
−1.6
−2.4

```
df = df_fair

alpha = 0.05
a = df.affairs.quantile(alpha/2)
b = df.affairs.quantile(1-alpha/2)

df = df[(df.affairs >= a) & (df.affairs <= b)]

df_fair_middle95 = df


df = df_fair
smf.ols(formula, df).fit().summary()
smf.ols(formula, df_fair_middle95).fit().summary()
smf.quantreg(formula, df).fit().summary()
smf.rlm(formula, df).fit().summary()
```


subway $
pizza $


subway $
pizza $
inflation


subway $
inflation

Correlation Does Not Imply Causation

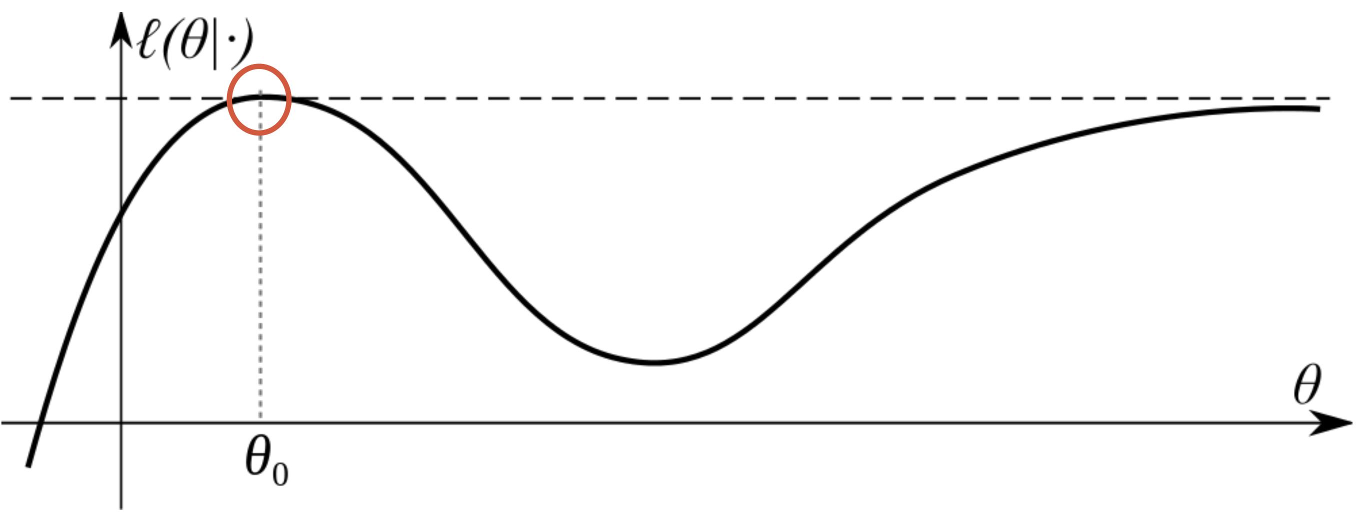
- $y \sim x :=$ "y has association with x"
- $y \leftarrow x :=$ "y because x"
- $y \sim x$ may be:
  - $y \leftarrow x$
  - $y \rightarrow x$
  - $z \rightarrow y \wedge z \rightarrow x$
- So $y \sim x$ doesn't implies $y \leftarrow x$.
- *A good research design* $\wedge\ y \sim x$ can implies $y \leftarrow x$.

Suggested Wording

- Causation
  - “**Affect**” “Influence”
  - “Predict” “Prediction”, in regression analysis especially.
- Correlation
  - “Associate” “**Association**”
  - “Correlate” “Correlation”, in correlation analysis especially.

More Models

- Discrete Models, like Logit Model:
  - $y \in \{0, 1\}$
- Mixed Model for estimate both group and subject effect:
  - $y = X\beta + Zu + \varepsilon$
- Time Series Models, like Autoregressive Model:
  - $x_t = c + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \ldots + \varphi_p x_{t-p} + \varepsilon_t$
- Check all the models that StatsModels supports.

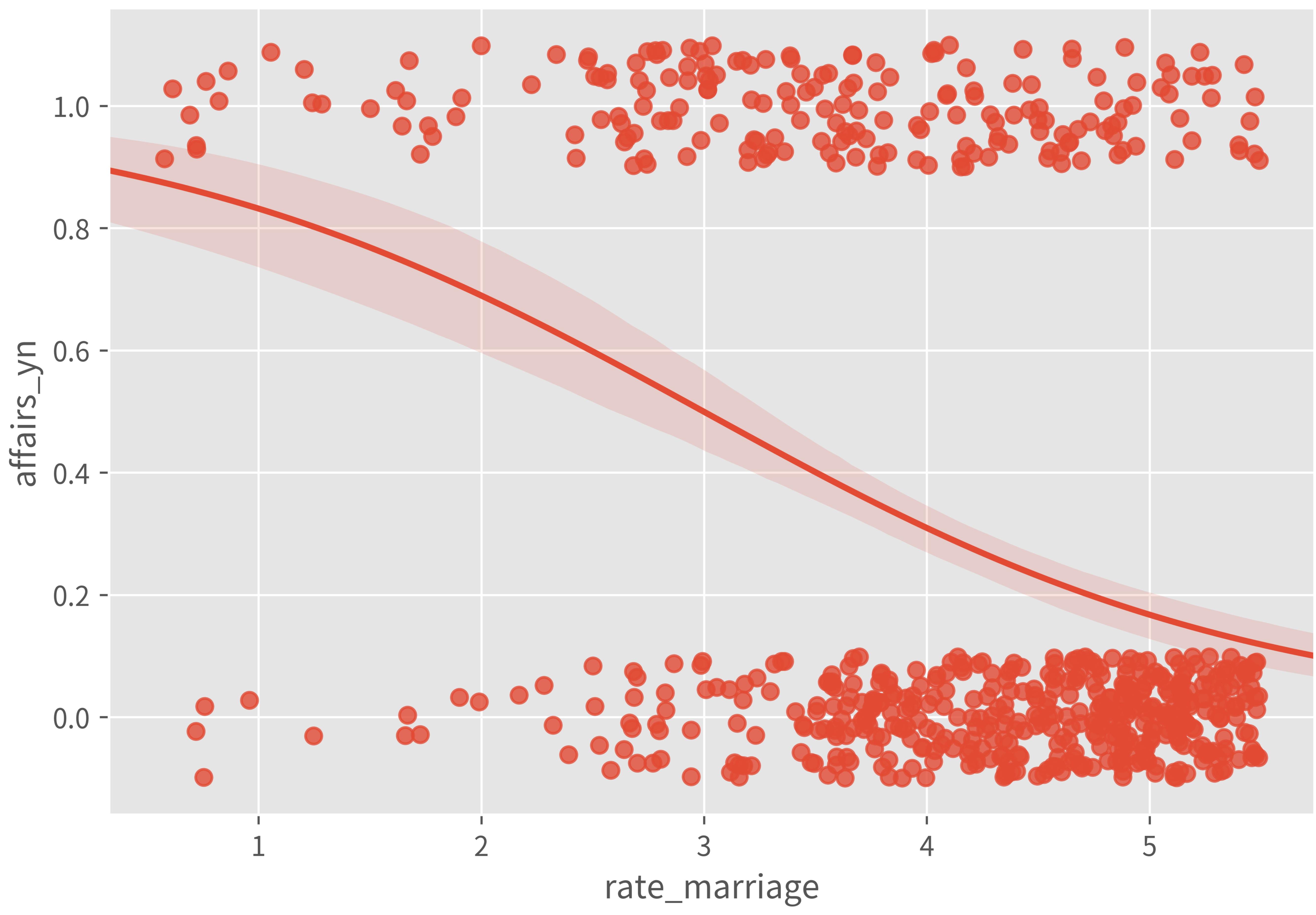


More Estimations

- MLE, Maximum Likelihood Estimation.

  ← Usually find by numerical methods.

- TSLS, Two-Stage Least Squares.
  - $y \leftarrow (x \leftarrow z)$
  - Handle the endogeneity: $E[\varepsilon | X] \neq 0$.



1.0
0.8
0.6
0.4
0.2
0.0
affairs_yn
1
2
3
4
5
rate_marriage

Logit Regression Results

Dep. Variable:	affairs_yn	**No. Observations:**	6366
Model:	Logit	**Df Residuals:**	6346
Method:	MLE	**Df Model:**	19
Date:	Mon, 29 Apr 2019	**Pseudo R-squ.:**	0.1003
Time:	22:48:50	**Log-Likelihood:**	-3601.0
converged:	True	**LL-Null:**	-4002.5
		LLR p-value:	4.598e-158

	coef	std err	z	P>|z|	[0.025	0.975]
Intercept	0.6931	0.500	1.386	0.166	-0.287	1.673
C(rate_marriage)[T.2.0]	0.2231	0.581	0.384	0.701	-0.915	1.362
C(rate_marriage)[T.3.0]	-0.2122	0.523	-0.405	0.685	-1.238	0.813
C(rate_marriage)[T.4.0]	-1.2009	0.512	-2.345	0.019	-2.205	-0.197
C(rate_marriage)[T.5.0]	-1.6664	0.512	-3.256	0.001	-2.669	-0.663
C(rate_marriage)[1.0]:C(religious)[T.2.0]	0.5596	0.641	0.873	0.383	-0.696	1.816
C(rate_marriage)[2.0]:C(religious)[T.2.0]	-0.1398	0.345	-0.405	0.686	-0.817	0.537
C(rate_marriage)[3.0]:C(religious)[T.2.0]	-0.2657	0.184	-1.443	0.149	-0.626	0.095
C(rate_marriage)[4.0]:C(religious)[T.2.0]	-0.0294	0.132	-0.222	0.824	-0.288	0.230
C(rate_marriage)[5.0]:C(religious)[T.2.0]	-0.4791	0.140	-3.427	0.001	-0.753	-0.205
C(rate_marriage)[1.0]:C(religious)[T.3.0]	0.4769	0.629	0.758	0.448	-0.756	1.710
C(rate_marriage)[2.0]:C(religious)[T.3.0]	-0.5656	0.349	-1.622	0.105	-1.249	0.118
C(rate_marriage)[3.0]:C(religious)[T.3.0]	-0.3412	0.188	-1.811	0.070	-0.710	0.028
C(rate_marriage)[4.0]:C(religious)[T.3.0]	-0.4342	0.134	-3.239	0.001	-0.697	-0.171
C(rate_marriage)[5.0]:C(religious)[T.3.0]	-0.6133	0.137	-4.487	0.000	-0.881	-0.345
C(rate_marriage)[1.0]:C(religious)[T.4.0]	0.2231	0.975	0.229	0.819	-1.687	2.133
C(rate_marriage)[2.0]:C(religious)[T.4.0]	-1.3218	0.504	-2.622	0.009	-2.310	-0.334
C(rate_marriage)[3.0]:C(religious)[T.4.0]	-0.7105	0.286	-2.486	0.013	-1.271	-0.150
C(rate_marriage)[4.0]:C(religious)[T.4.0]	-0.7732	0.210	-3.675	0.000	-1.186	-0.361
C(rate_marriage)[5.0]:C(religious)[T.4.0]	-1.3503	0.212	-6.355	0.000	-1.767	-0.934

Logit Model

- The coef is log-odds.
- Use $\exp(x)/(\exp(x)+1)$ to transform back to probability:
  - 0.6931 → 67%
  - " - 1.6664 → 27%
  - " - 1.3503 → 9%
- Or:
  - `.predict(dict(rate_marriage=[1, 5, 5], religious=[1, 1, 4]))`

```python
df = df_fair
df = df.assign(affairs_yn=(df.affairs > 0).astype(float))
df_fair_2 = df
```

```python
df = df_fair_2.sample(frac=0.1, random_state=20190429)
sns.regplot(data=df, x='rate_marriage', y='affairs_yn',
            logistic=True,
            x_jitter=1/2, y_jitter=0.2/2)
```

```python
df = df_fair_2
(smf
 .logit('affairs_yn'
        '~ C(rate_marriage)'
        '+ C(rate_marriage):C(religious)', df)
 .fit()
 .summary())
```

Recap

- Choose the method by the assumption.
- Correlation Analysis: Gets an overview.
- Understand Regression Result:
  - **Plotting**, Adj. R-squared, Cond. No., Durbin-Watson, etc.
- Model Specification Using the R Formula:
  - `y ~ 0 + x`
  - `y ~ x*z`
  - `y ~ x:z`
- Covariance Types: Use HC3 by default.
- Let's explain and predict efficiently! 📈