REGRESSION PART 4: DUMMY AND ONE-HOT CODING

Hsin-Min Lu
盧信銘
台大資管系

Qualitative Predictors

- How do you stick “men” and “women” (category listings) into a regression equation?

- Code them as indicator variables (dummy variables)

- For example we can “code” Males=0 and Females= 1.

Dummy Coding

- Suppose we want to include income and gender.
- Two genders (male and female). Let

$$\boxed{\text{Gender}_i = \begin{cases} 0 \text{ if male} \\ 1 \text{ if female} \end{cases}}$$

- then the regression equation is

$$\boxed{Y_i \approx \beta_0 + \beta_1 \text{Income}_i + \beta_2 Gender_i = \begin{cases} \beta_0 + \beta_1 \text{Income}_i \text{ if male} \\ \beta_0 + \beta_1 \text{Income}_i + \beta_2 \text{ if female} \end{cases}}$$

- β_2 is the average extra balance each month that females have for given income level. Males are the “baseline”.

Regression coefficients

	Coefficient	Std Err	t-value	p-value
Constant	233.7663	39.5322	5.9133	0.0000
Income	0.0061	0.0006	10.4372	0.0000
Gender_Female	24.3108	40.8470	0.5952	0.5521

Deviation Coding

- There are different ways to code categorical variables.
- Two genders (male and female). Let

$$\boxed{Gender_i = \begin{cases} -1 \text{ if male} \\ 1 \text{ if female} \end{cases}}$$

- then the regression equation is

$$\boxed{Y_i \approx \beta_0 + \beta_1 \text{Income}_i + \beta_2 Gender_i = \begin{cases} \beta_0 + \beta_1 \text{Income}_i - \beta_2, \text{ if male} \\ \beta_0 + \beta_1 \text{Income}_i + \beta_2, \text{ if female} \end{cases}}$$

- β_2 is the average amount that females are above the average, for any given income level. β_2 is also the average amount that males are below the average, for any given income level.

Coding Scheme Selection

- Different coding scheme may give you different parameter estimation result for the same dataset.
- If you are handling experiment data, need to follow the traditional wisdom of statistical analysis.
- However, in a more “data mining” flavor situation, we usually **use dummy coding**.
- The main reason is because we care about the effect of an individual feature taking a specific value, i.e., the “simple effect.”

Dummy Coding (Cont'd.)

- In dummy coding, if a feature can take K possible values, then we should use K-1 dummy variables.
- What if we use K dummy variables?
  - Introduce linear dependency in feature matrix.
- Ordinary Least Square (OLS) required that the feature matrix to be full-rank.
  - Need to remove features that cause linear dependency.
  - QR factorization can handle this situation well.
  - Another way to handle the situation is through **regularization**. (will discuss later)

Dummy Coding and One-Hot Encoding

- Dummy coding is a more “statistical” flavored term.
- Machine learning people use “one-hot” encoding.
- Main difference?
- If a feature can take K possible values,
  - Dummy coding: use K-1 dummy variables.
  - One-Hot encoding: use K binary variables.

Example: Dummy vs. One-Hot (aka One-of-K)

- Example: a variable “fruit” can take three possible values {apple, orange, grape}
- Suppose three data points have fruit = apple, orange, grape, grape.
- Dummy coding: [1 0], [0 1], [0 0], [0 0]
- One-hot: [1 0 0], [0 1 0], [0 0 1], [0 0 1]

- Note the linear dependent problem in one-hot encoding.
- Can use regularization to handle this problem in linear regression.

Pre-Selecting Feature Values (for a Given Variable)

- When converting input features to dummy coding (or one-hot encoding), we inflated number of input features.
- When a variable can take a lot of values, then brute force dummy coding is a bad idea.
- We should only “pre-select” meaningful values to code.

- Two strategy to select feature values:
- 1: (Document) frequency
- 2: Feature filtering (i.e., via t-value).

Dummy Coding (Cont'd.)

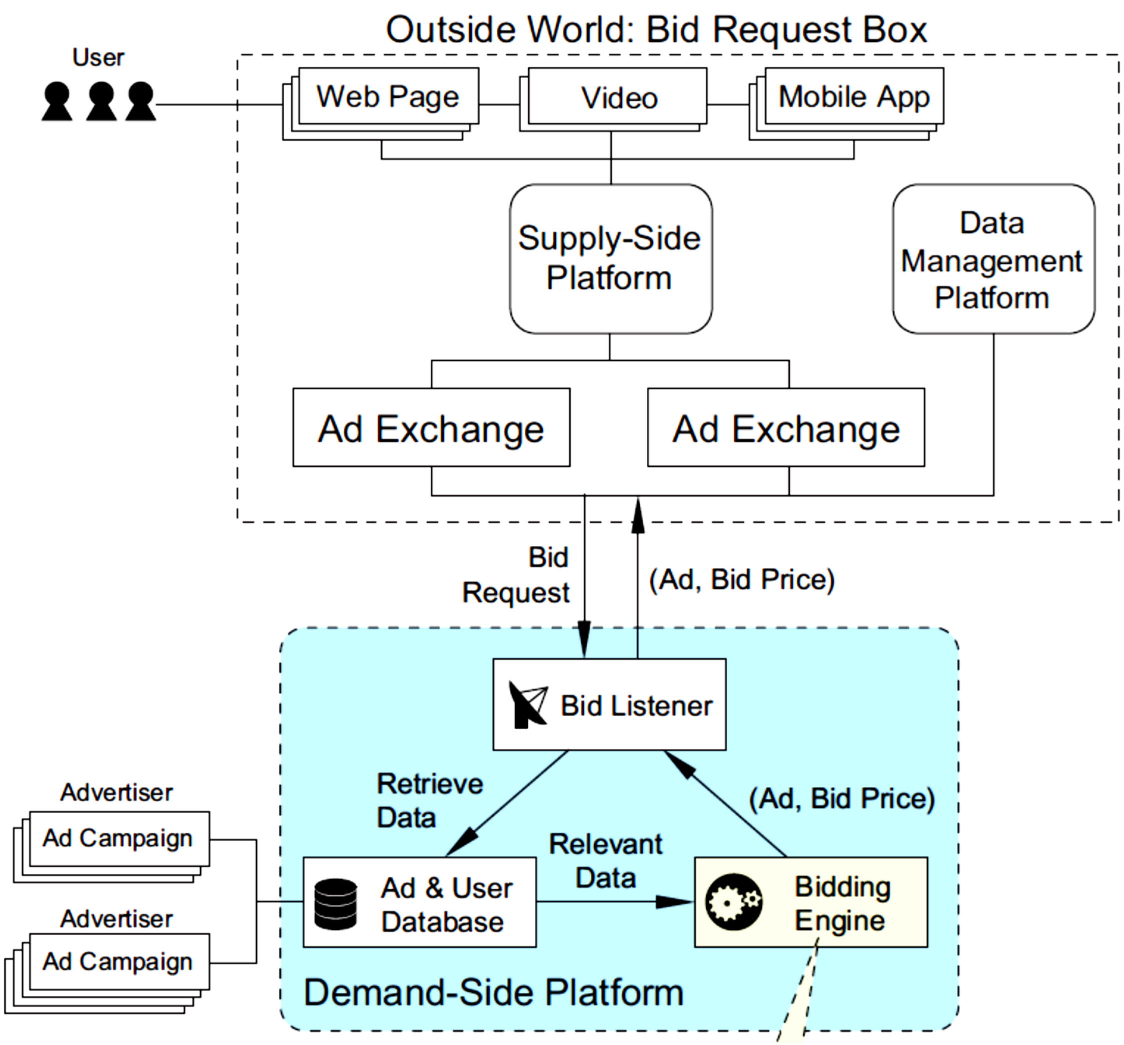
- 1: (Document) frequency
- Do not select features that appear only 1 time in the training dataset. (Why?)
- Need to determine a threshold that filter low frequency features.
  - >=2 is a conservative setting.
  - Sometime we use more aggressive setting (e.g. >=5) to reduce the number of dummy variables.
- Similarly, do not code a feature value that appear in all rows. (why?)

Dummy Coding (Cont'd.)

- 2: Feature filtering and ranking (i.e., via t-value).
- Compute the t-value of regression coefficients for individual feature value, then rank by abs(t-value).
- Select only top M features to be included by adjusted R-square.
- Feature matrix X = $\begin{bmatrix} 1 & x_{i1} \\ 1 & x_{i2} \\ \vdots & \vdots \\ 1 & x_{in-1} \\ 1 & x_{in} \end{bmatrix}$, $\hat{\beta} = (X^T X)^{-1} X^T Y$,
- $\hat{\Sigma} = Cov(\hat{\beta}) = (X^T X)^{-1} \hat{\sigma}^2$
- $t - value = \hat{\beta}_1 / \sqrt{\hat{\Sigma}_{11}}$ (parameter index start with 0)
- What is the downslide of this approach?

Example: Real Time Bidding (RTB)

- RTB allows advertisers (廣告商; the demand side) to bid on a display ad impression in real time when it is being generated.
- A common goal is to maximize direct visit or conversion.



RTB: Winning Price Prediction

- When a AD slot appear, the advertiser need to determine how much it is going to bid on that slot.
- There are usually multiple bids for a slot, and the highest bidder win.
- The winner is going to pay the highest losing price (i.e., the winning price).
  - The “second price auction.”
- There are many issues to consider. For example, we do not want to bid on every ad slot. Instead, we are going to focus on ad slots that will generate conversion.
- If we do this right, the other issue is to predict the winning price so that we know how much to bid on the slot.

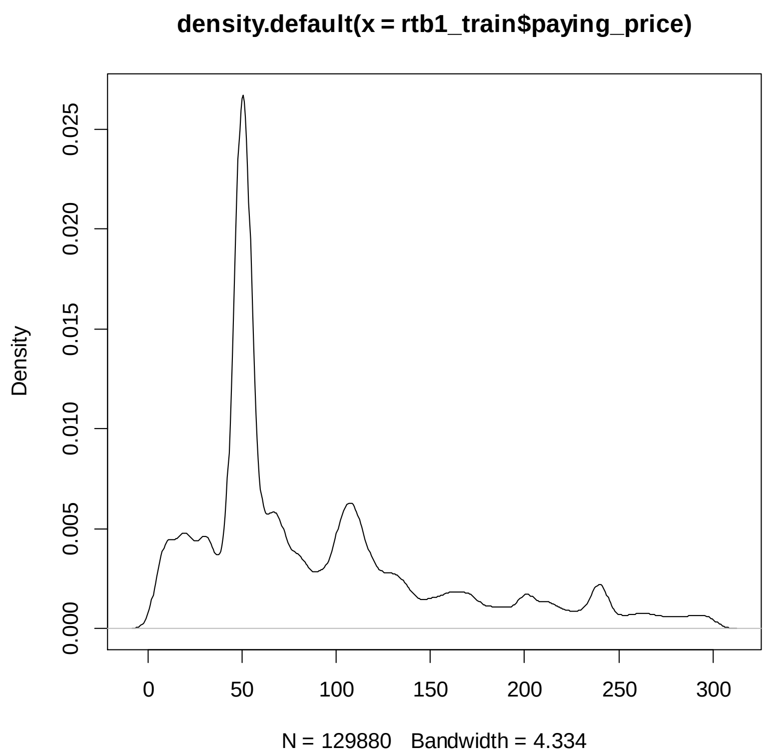
Data

- We are going to predict "paying price" using IP.
- IP is the user IP address.

SN	Column	Example
*1	Bid ID	01530000008a77e7ac18823f5a4f5121
2	Timestamp	20130218134701883
3	Log Type	1
*4	iPinYou ID	35605620124122340227135
5	User-Agent	Mozilla/5.0 (compatible; MSIE 9.0; \ Windows NT 6.1; WOW64; Trident/5.0)
*6	IP	118.81.189.*
7	Region ID	15
8	City ID	16
9	Ad Exchange	2
*10	Domain	e80f4ec7f5bfbc9ca416a8c01cd1a049
*11	URL	hz55b000008e5a94ac18823d6f275121
12	Anonymous URL	null
13	Ad Slot ID	2147689_8764813
14	Ad Slot Width	300
15	Ad Slot Height	250
16	Ad Slot Visibility	SecondView
17	Ad Slot Format	Fixed
18	Ad Slot Floor Price	0
19	Creative ID	e39e178ffdf366606f8cab791ee56bcd
*20	Bidding Price	753
*21	Paying Price	15
*22	Landing Page URL	a8be178ffdf366606f8cab791ee56bcd
23	Advertiser ID	3358
*24	User Profile IDs	123,5678,3456

Exploring the Dataset

- Before doing anything, need to explore the dataset.
- Paying price summary statistics
- Min. 1st Qu. Median Mean 3rd Qu. Max.
- 4.00 50.00 63.00 90.32 118.00 300.00
- Kernel density of paying price:
- How do you describe the outcome variable?
  - Multiple mode: around 65, 120, 240,
  - Some winning price quite low



Exploring the Dataset (Cont'd.)

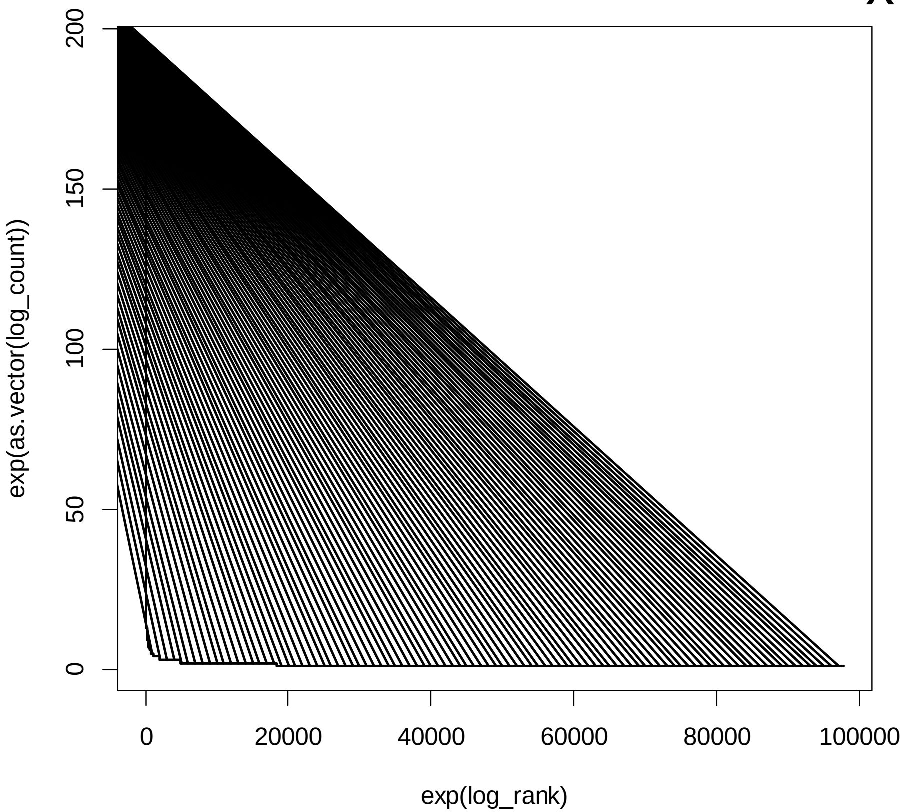
- IP Address frequency distribution:
- 124.165.213.* 125.39.238.* 202.107.244.* 118.122.85.* 61.185.133.*
- 193 161 107 102 82
- 219.139.102.* 112.90.90.* 182.148.111.* 120.128.7.* 210.38.1.*
- 78 73 71 67 67

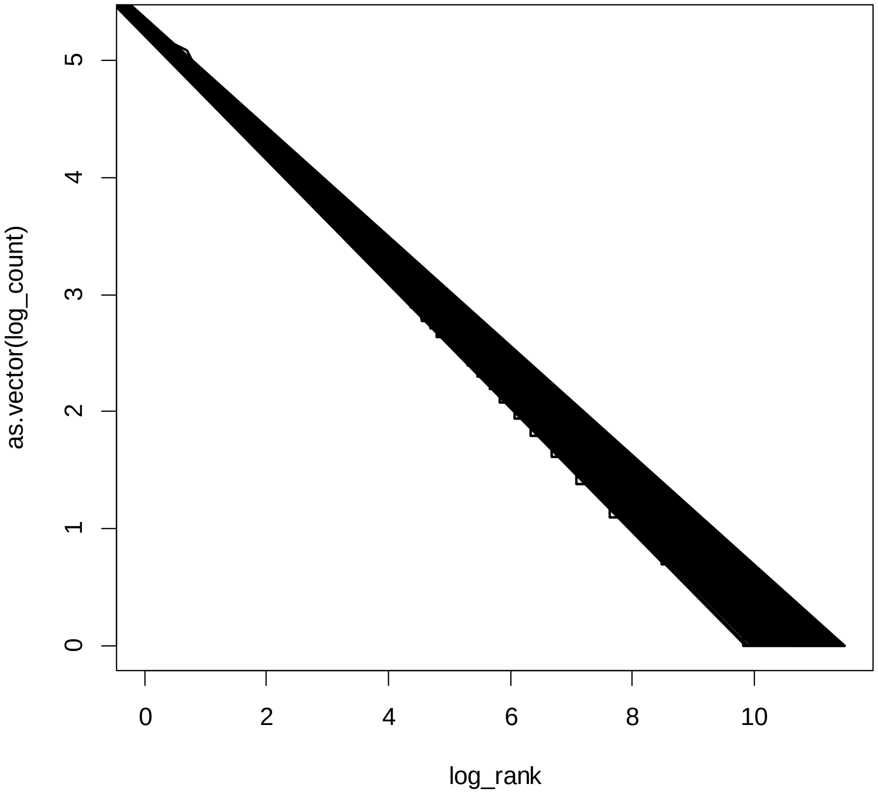
Frequency Distribution

- Not transformed
- x-axis: rank; y-axis: count
-

Log-Log Scale

x-axis: log-rank; y-axis: log-count





Frequency Distribution

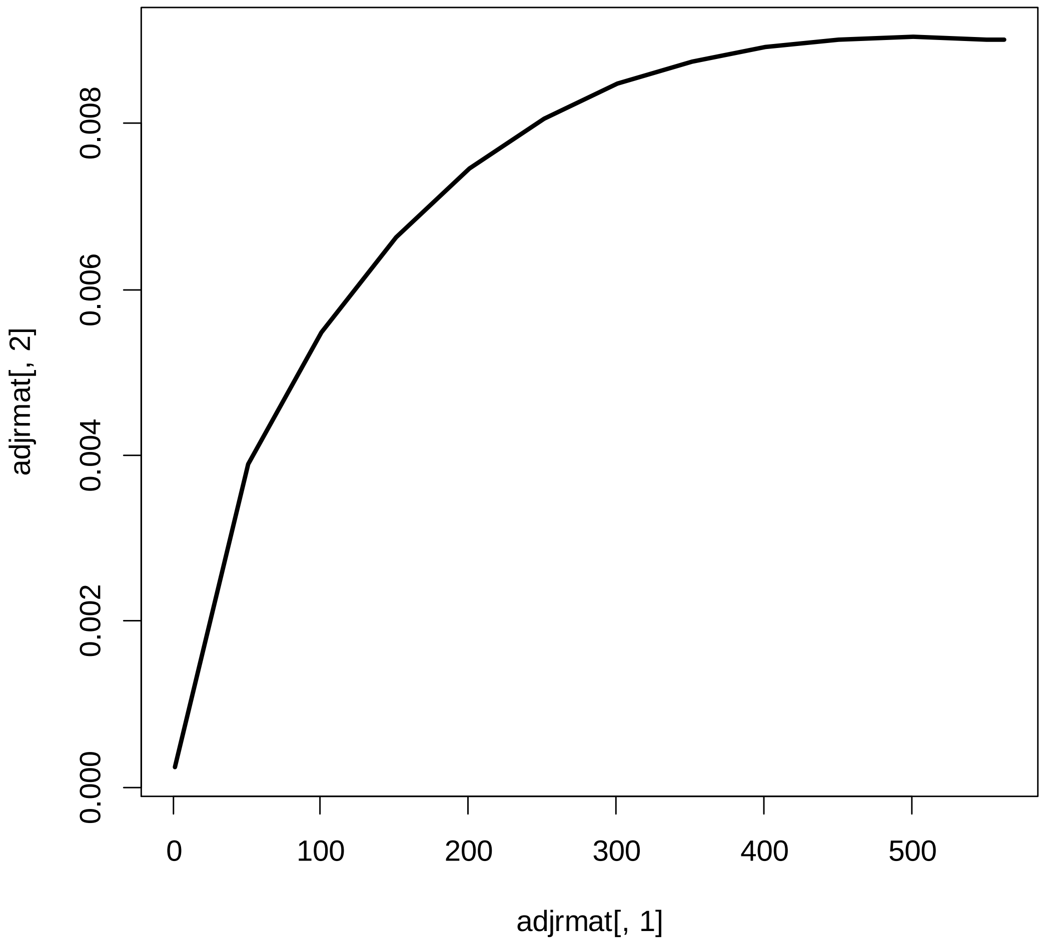
- Log-log scale plot is close to a straight line
- Commonly referred as the “long tail” effect.
- Meaning: Cutting by frequency can greatly reduce the number of unique values.
- For IP address:
  - Number of unique values = 97,883
  - Threshold=2, # of unique values = 18,387
  - Threshold=3, # of unique values = 4,852
  - Threshold=5, # of unique values = 1,195
- We are going to use threshold=5, which gives us 1,195 unique values.

Filtering features via t-value

- For each IP-address, compute its t-value
- Sort features by the absolute value of t-value (from large to small).
- Larger absolute t-value means more important.
- Select top k features to be used in subsequent analysis

Note the Trend of Adj-R2

- > adjrmat
-       [,1]   [,2]
- [1,]   1 0.0002404851
- [2,]   51 0.0038870548
- [3,] 101 0.0054895772
- [4,] 151 0.0066317499
- [5,] 201 0.0074576135
- [6,] 251 0.0080520593
- [7,] 301 0.0084724273
- [8,] 351 0.0087528488
- [9,] 401 0.0089204734
- [10,] 451 0.0090162759
- [11,] 501 0.0090476604
- [12,] 551 0.0090181959
- [13,] 563 0.0090024392



Abs(t-value) Threshold

- If the goal is to maximize Adj R2, then we should select features with abs(t-value)>=1.
- It can be shown that including a feature with abs(t-value) >=1 (with all other features considered) will increase the model adjusted R2.
- However, we only have t-value using that feature along.
- If feature correlation is not a serious problem, than the “individual” t-value is a good proxy for its t-value when all other features were included.
- To be conservative, consider all features with abs(t-value)>=0.9.

Final Regression Result

```
Coefficients:
               Estimate Std. Error t value                 Pr(>|t|)
(Intercept)     90.4686     0.1891 478.432 < 0.0000000000000002 ***
ip_1            27.1583     4.7946   5.664         0.0000000148 ***
ip_2            28.8171     5.2488   5.490         0.0000000402 ***
ip_3           -34.1157     6.5928  -5.175         0.0000002287 ***
ip_4           -35.8020     8.3875  -4.268         0.0000196955 ***
ip_5           -27.4032     6.4371  -4.257         0.0000207261 ***
ip_6           -33.6115     8.3875  -4.007         0.0000614474 ***
ip_7           -29.2369     7.3524  -3.976         0.0000699757 ***
ip_8           105.0314    27.1725   3.865             0.000111 ***
ip_9           112.1314    29.7658   3.767             0.000165 ***
ip_10          100.5314    27.1725   3.700             0.000216 ***
ip_11          -39.1186    10.5253  -3.717             0.000202 ***
...
```

Final Regression Result

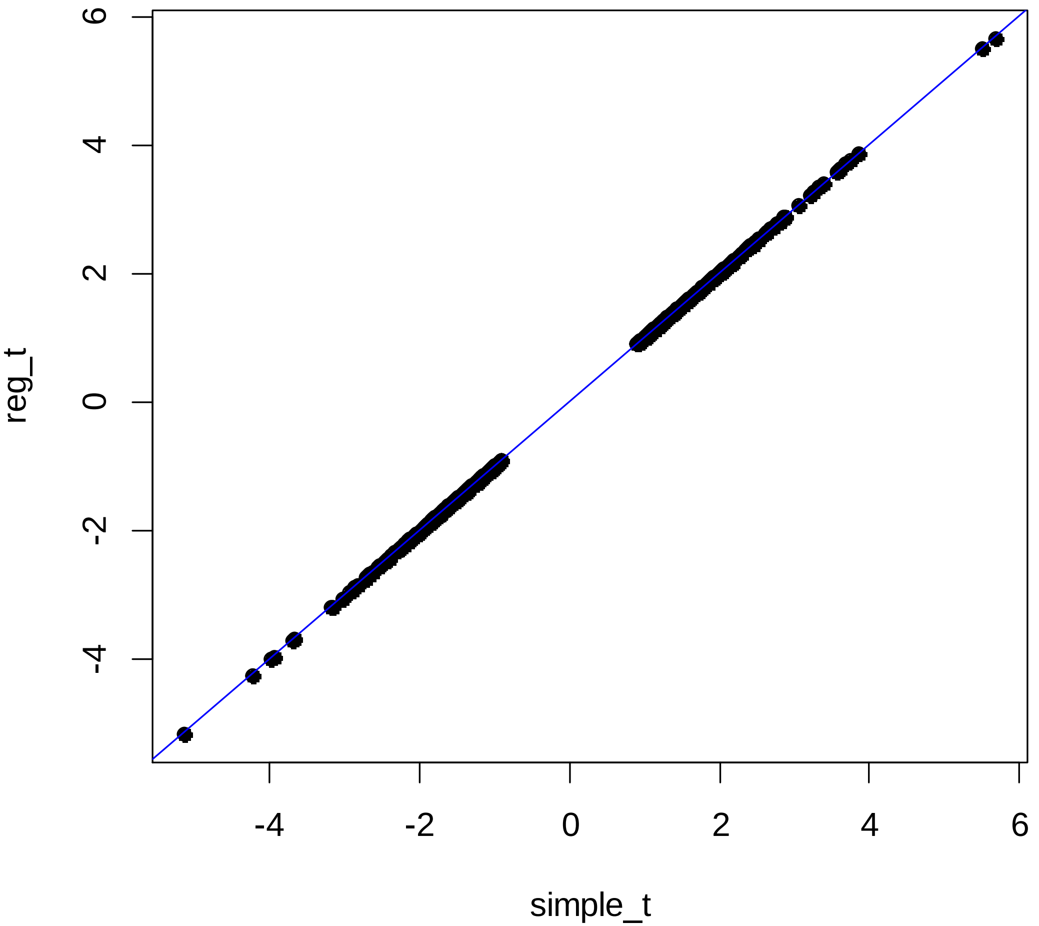
```
...
ip_496      -30.2686    29.7658  -1.017            0.309207
ip_497      -30.2686    29.7658  -1.017            0.309207
ip_498      -27.6353    27.1725  -1.017            0.309140
ip_499       29.9314    29.7658   1.006            0.314628
ip_500       29.9314    29.7658   1.006            0.314628
ip_501       25.2457    25.1569   1.004            0.315608
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 66.56 on 129378 degrees of freedom
Multiple R-squared:  0.01287,   Adjusted R-squared:  0.009048
F-statistic: 3.367 on 501 and 129378 DF,  p-value: < 0.0000000000000022
```

How Good is the t-value from Simple Regression?

- X-axis: t-value from simple regression
- Y-axis: t-value from full regression

- t-value from simple regression is a very good approximation for their performance in the full model.



Summary of Dummy Coding Procedure

- Apply the following procedure for a variable that take qualitative values
- Compute the frequency distribution of realized values
  - K = # of unique values
- How does this frequency distribution looks like?
  - Only a few unique values (e.g., male, female; several cities)
  - ➔ Use K-1 dummy variables

  - There are a lot of unique values (like the IP example)
  - ➔ Adopt frequency threshold,
  - ➔compute t-value using simple regression,
  - ➔apply abs(t-value)≥1 threshold.