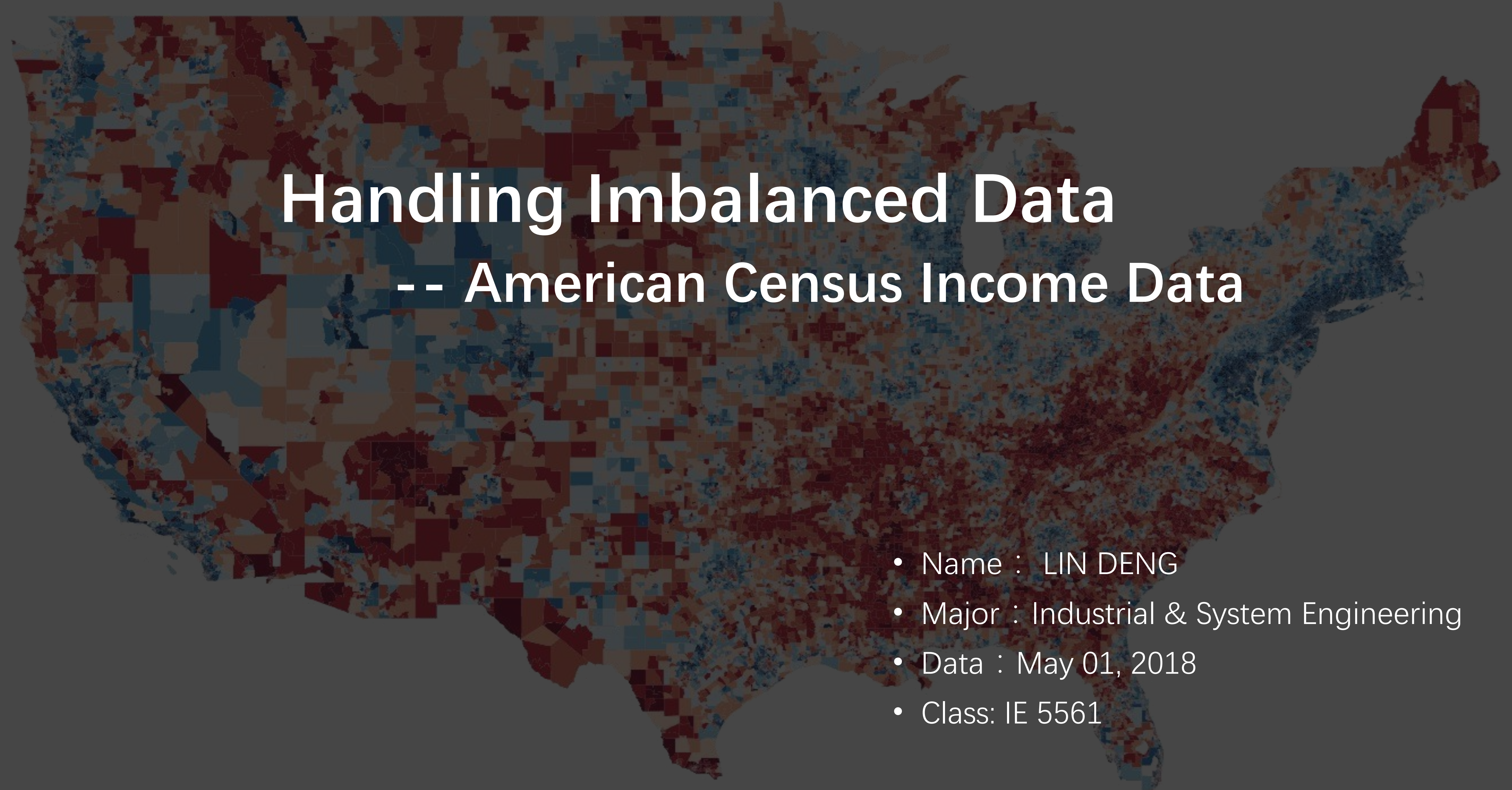


# Handling Imbalanced Data

## -- American Census Income Data

- Name： LIN DENG
- Major：Industrial & System Engineering
- Data：May 01, 2018
- Class: IE 5561

# CONTENTS

# CONTENTS

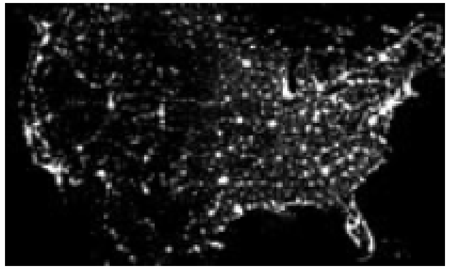
# CONTENTS

# Introduction & Background

## Census Income Data Set

*Download*: Data Folder, Data Set Description

**Abstract**: Predict whether income exceeds $50K/yr based on census data. Also known as "Adult" dataset.



| | | | | | |
|---|---|---|---|---|---|
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 48842 | **Area:** | Social |
| **Attribute Characteristics:** | Categorical, Integer | **Number of Attributes:** | 14 | **Date Donated** | 1996-05-01 |
| **Associated Tasks:** | Classification | **Missing Values?** | Yes | **Number of Web Hits:** | 258000 |

UCI Machine Learning Repository

United States™

Census

Bureau

# Introduction & Background

**training_data: 41 variables, 199,523 obs**
**test_data: 41 variables, 99,762 obs**

```
 91 distinct values for attribute #0 (age) continuous
  9 distinct values for attribute #1 (class of worker) nominal
 52 distinct values for attribute #2 (detailed industry recode) nominal
 47 distinct values for attribute #3 (detailed occupation recode) nominal
 17 distinct values for attribute #4 (education) nominal
240 distinct values for attribute #5 (wage per hour) continuous
  3 distinct values for attribute #6 (enroll in edu inst last wk) nominal
  7 distinct values for attribute #7 (marital stat) nominal
 24 distinct values for attribute #8 (major industry code) nominal
 15 distinct values for attribute #9 (major occupation code) nominal
  5 distinct values for attribute #10 (race) nominal
 10 distinct values for attribute #11 (hispanic origin) nominal
  2 distinct values for attribute #12 (sex) nominal
  3 distinct values for attribute #13 (member of a labor union) nominal
  6 distinct values for attribute #14 (reason for unemployment) nominal
  8 distinct values for attribute #15 (full or part time employment stat) nominal
132 distinct values for attribute #16 (capital gains) continuous
113 distinct values for attribute #17 (capital losses) continuous
```

# Introduction & Background

**training_data: 41 variables, 199,523 obs**
**test_data: 41 variables, 99,762 obs**

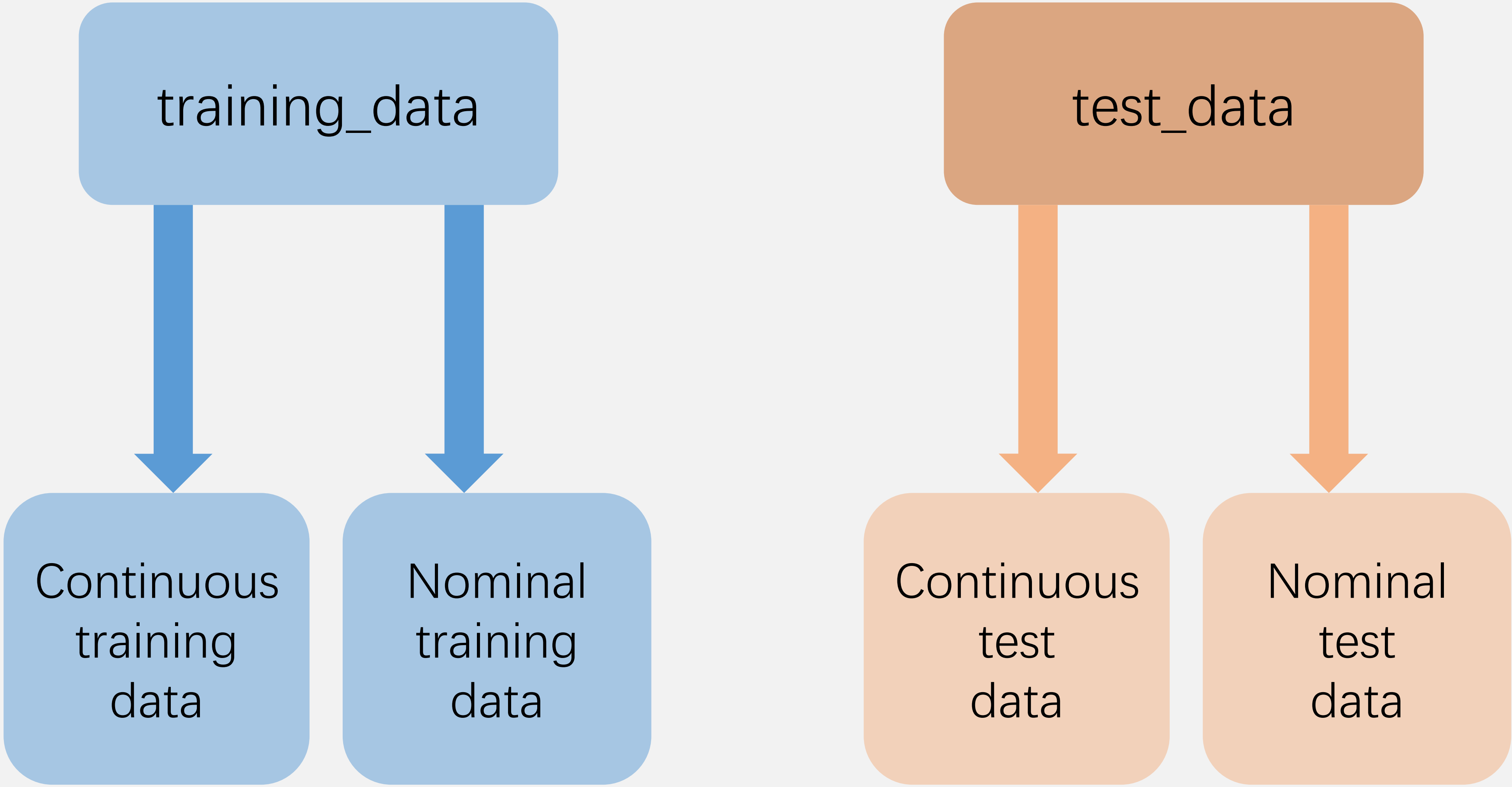
```
  91 distinct values for attribute #0 (age) continuous
   9 distinct values for attribute #1 (class of worker) nominal
  52 distinct values for attribute #2 (detailed industry recode) nominal
  47 distinct values for attribute #3 (detailed occupation recode) nominal
  17 distinct values for attribute #4 (education) nominal
1240 distinct values for attribute #5 (wage per hour) continuous
   3 distinct values for attribute #6 (enroll in edu inst last wk) nominal
   7 distinct values for attribute #7 (marital stat) nominal
  24 distinct values for attribute #8 (major industry code) nominal
  15 distinct values for attribute #9 (major occupation code) nominal
   5 distinct values for attribute #10 (race) nominal
  10 distinct values for attribute #11 (hispanic origin) nominal
   2 distinct values for attribute #12 (sex) nominal
   3 distinct values for attribute #13 (member of a labor union) nominal
   6 distinct values for attribute #14 (reason for unemployment) nominal
   8 distinct values for attribute #15 (full or part time employment stat) nominal
 132 distinct values for attribute #16 (capital gains) continuous
 113 distinct values for attribute #17 (capital losses) continuous
```

# II Data Separation & Basic Visualization

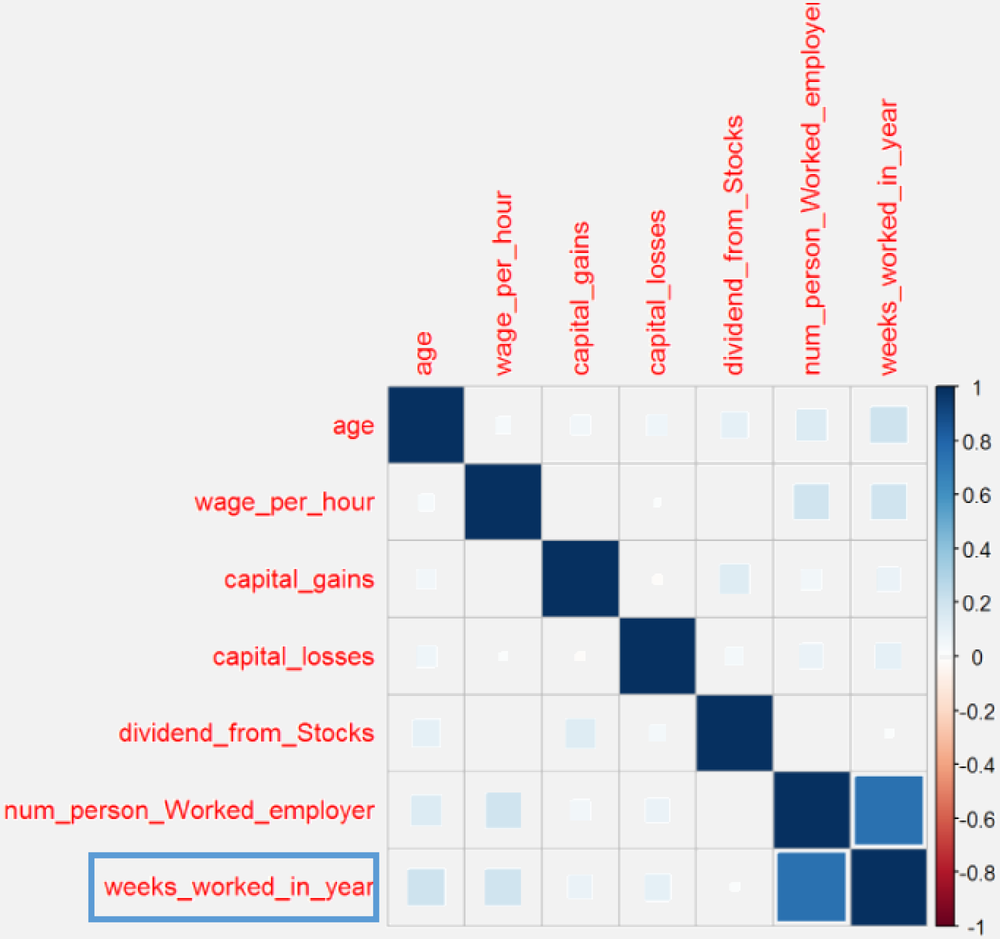
training_data

test_data

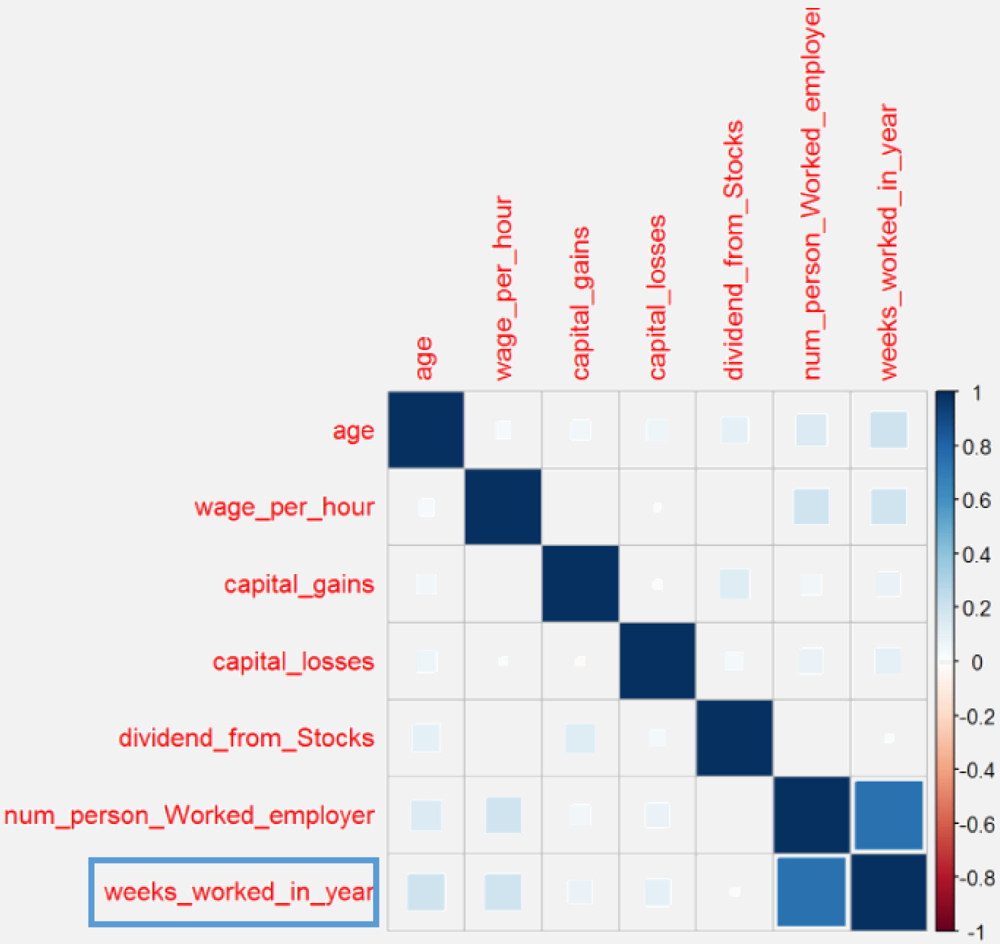
# II Data Separation & Basic Visualization

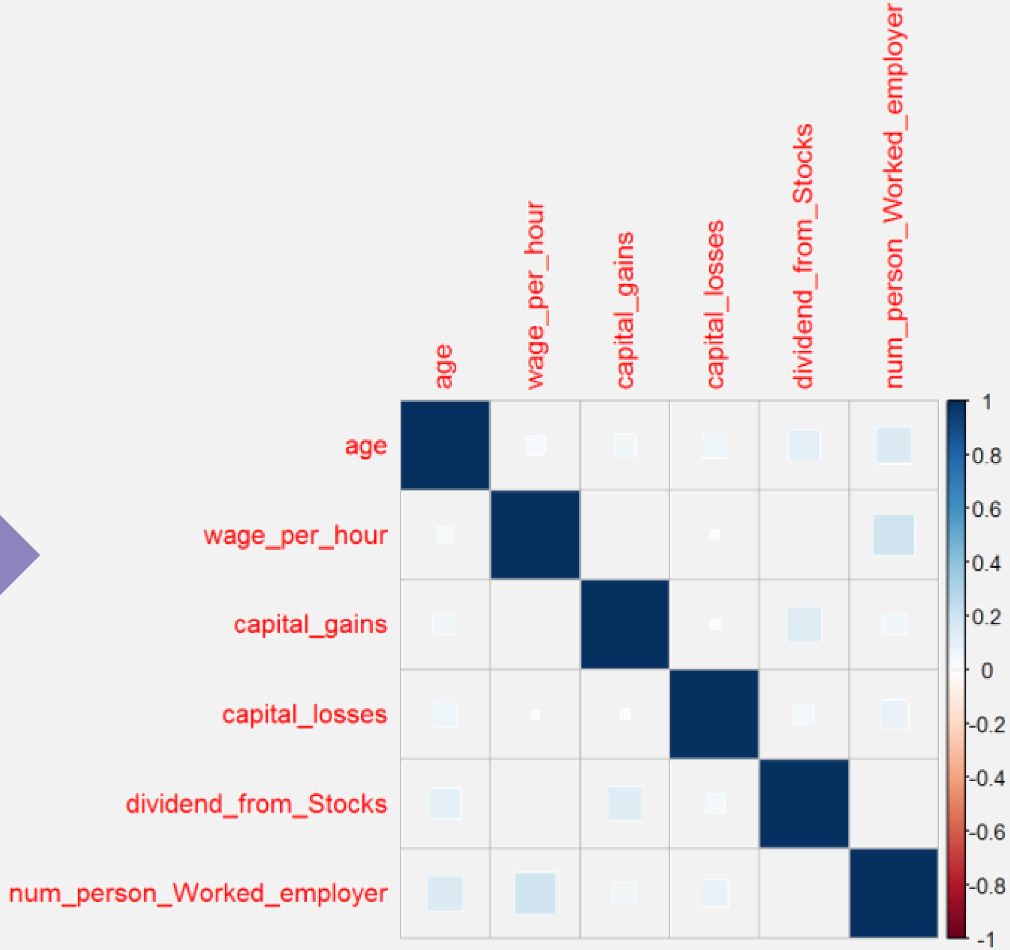


# II Data Separation & Basic Visualization

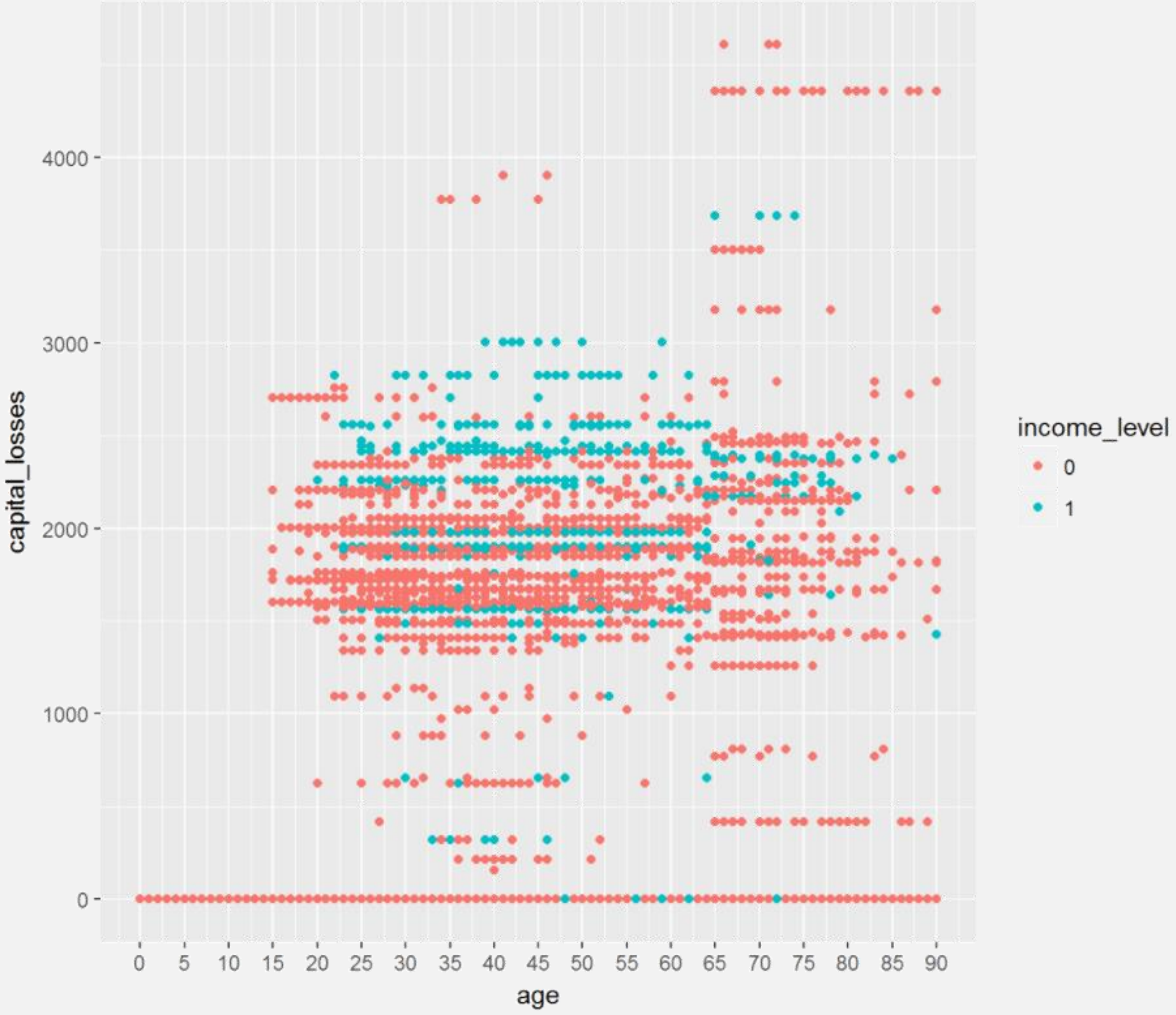


# II Data Separation & Basic Visualization



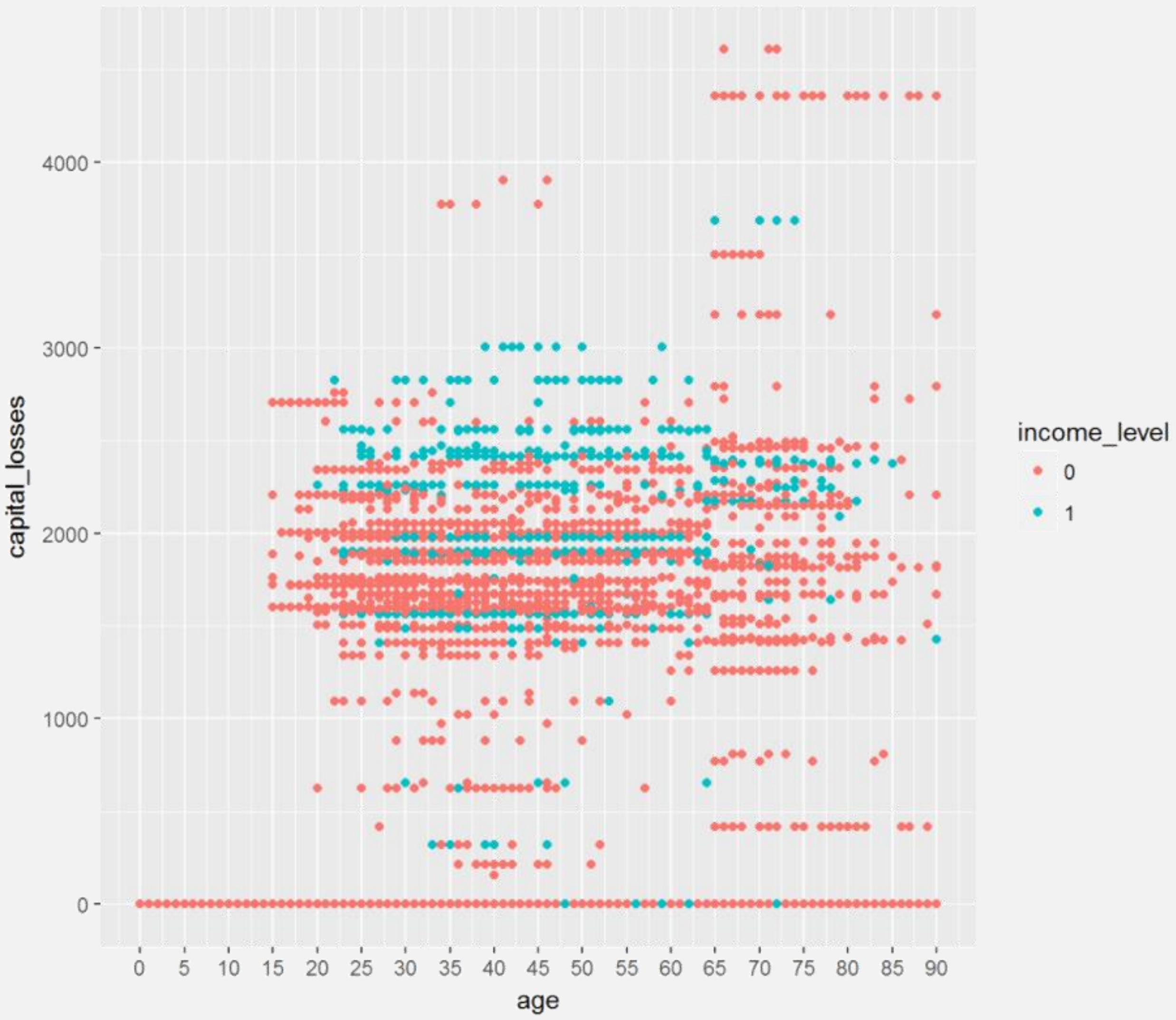



# II Data Separation & Basic Visualization

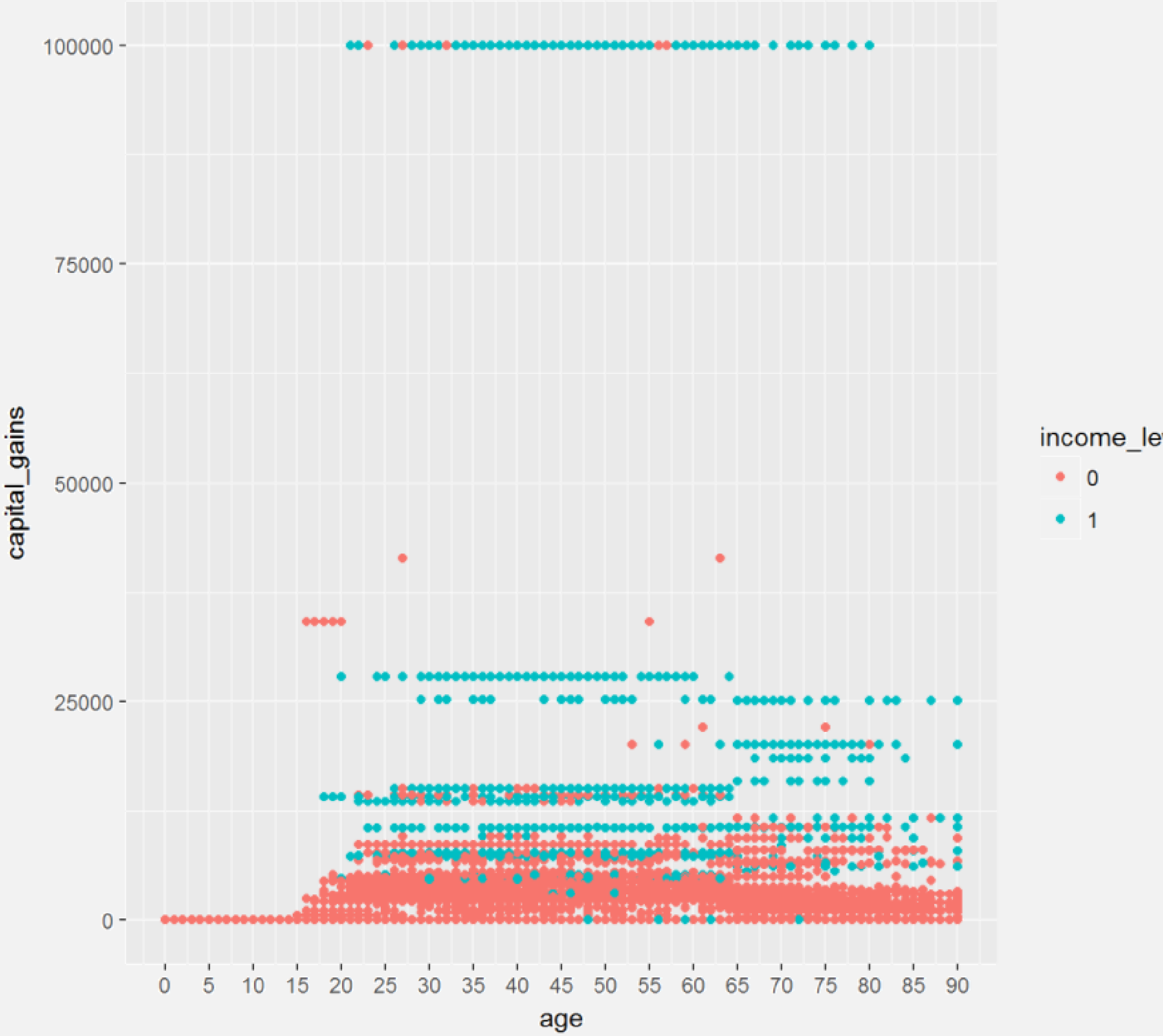



## Age ~ capital_losses

# II Data Separation & Basic Visualization

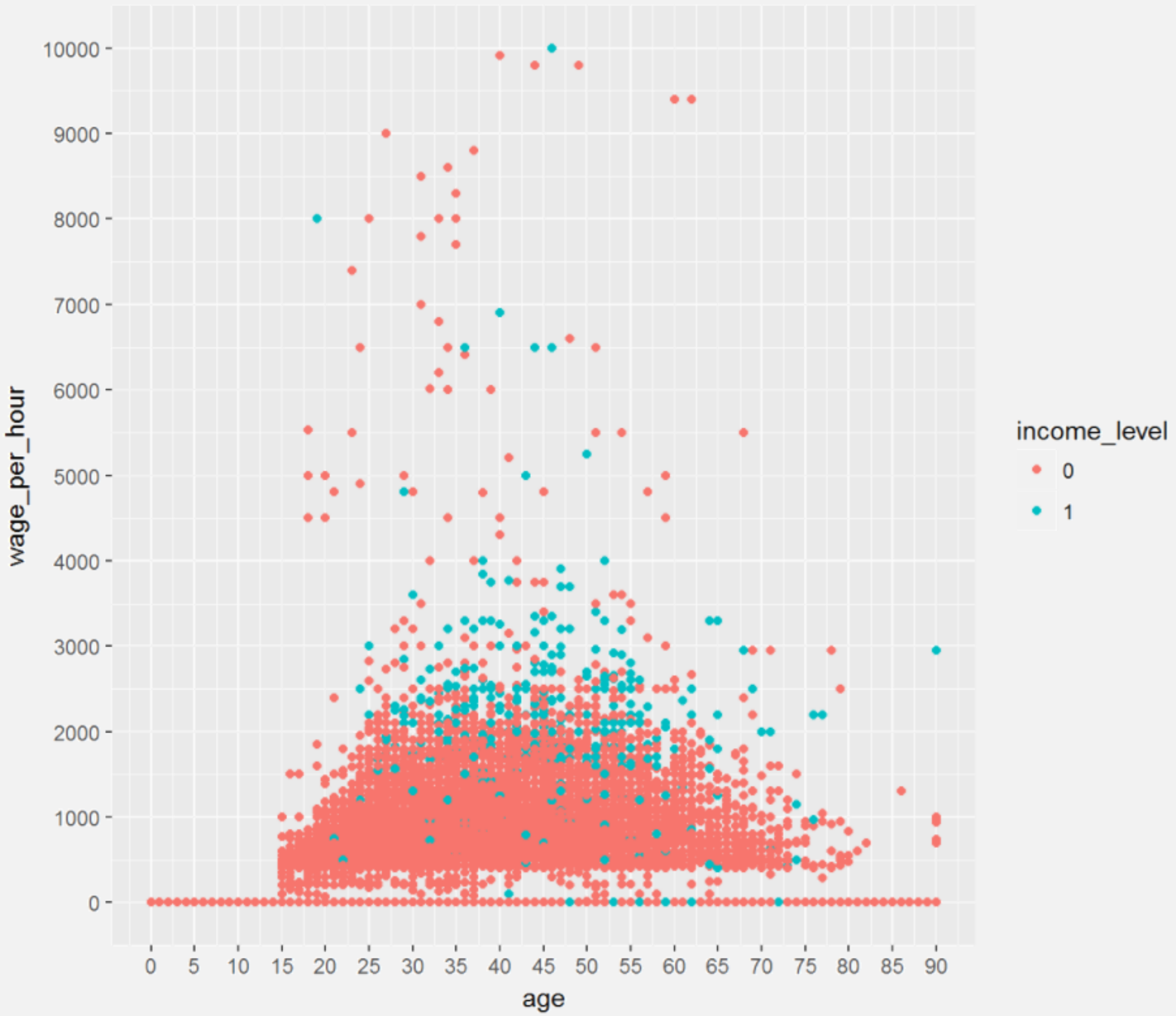



Age ~ capital_losses



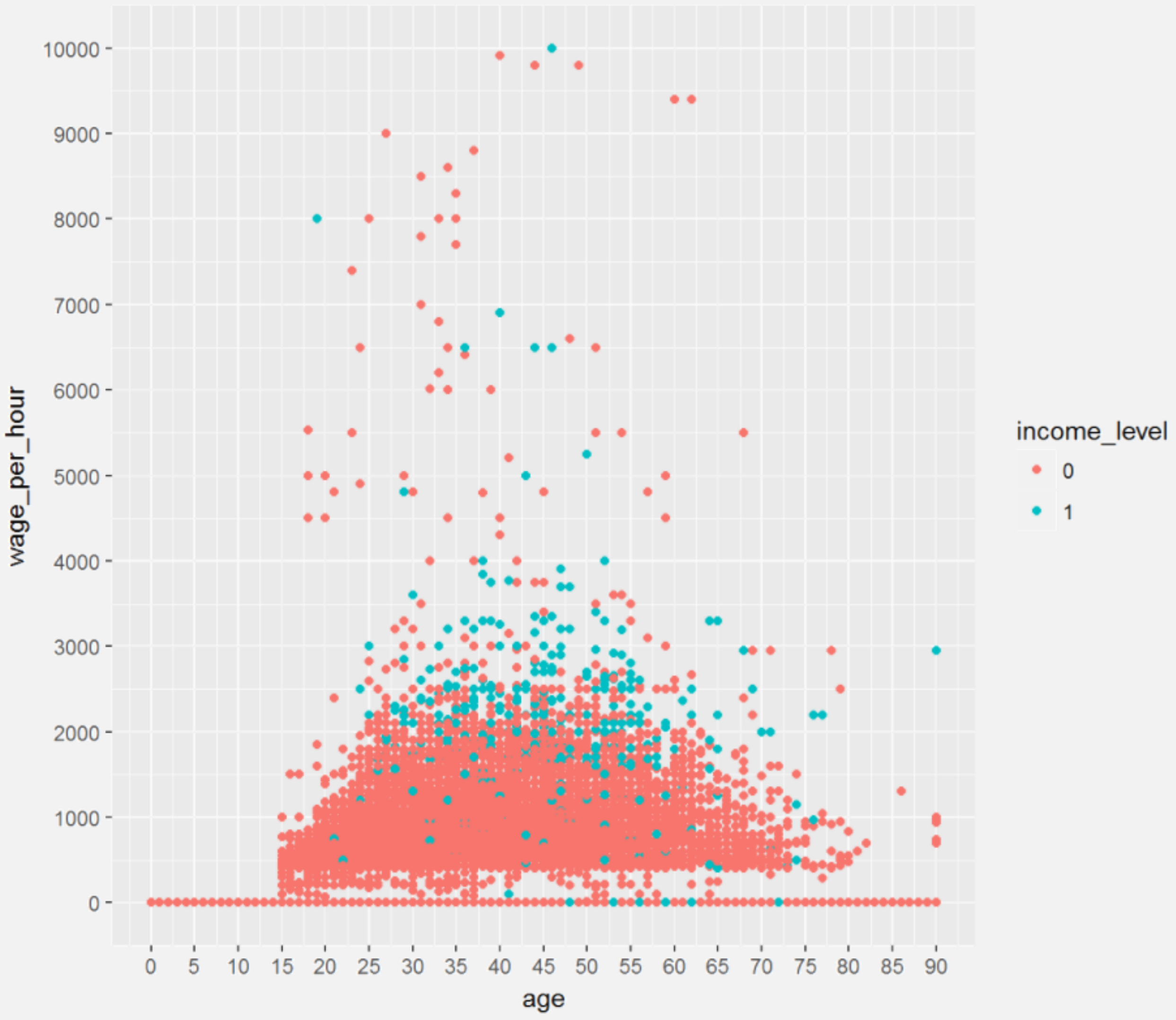

Age ~ capital_gains

# II Data Separation & Basic Visualization

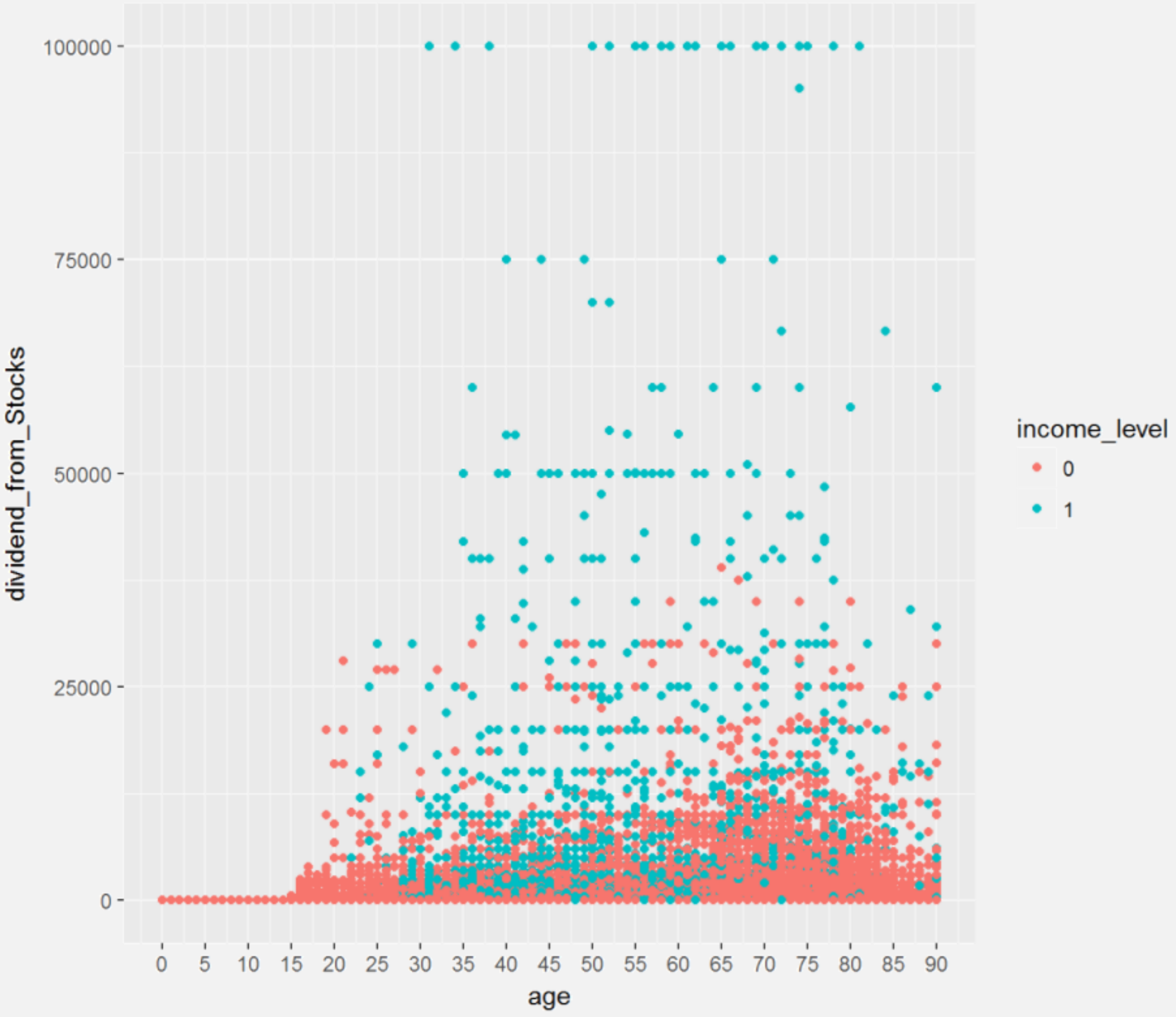



Age ~ Wage_per_hour

# II Data Separation & Basic Visualization




Age ~ Wage_per_hour




Age ~ dividend_from_stocks

# III Resample Data

Response: income_level

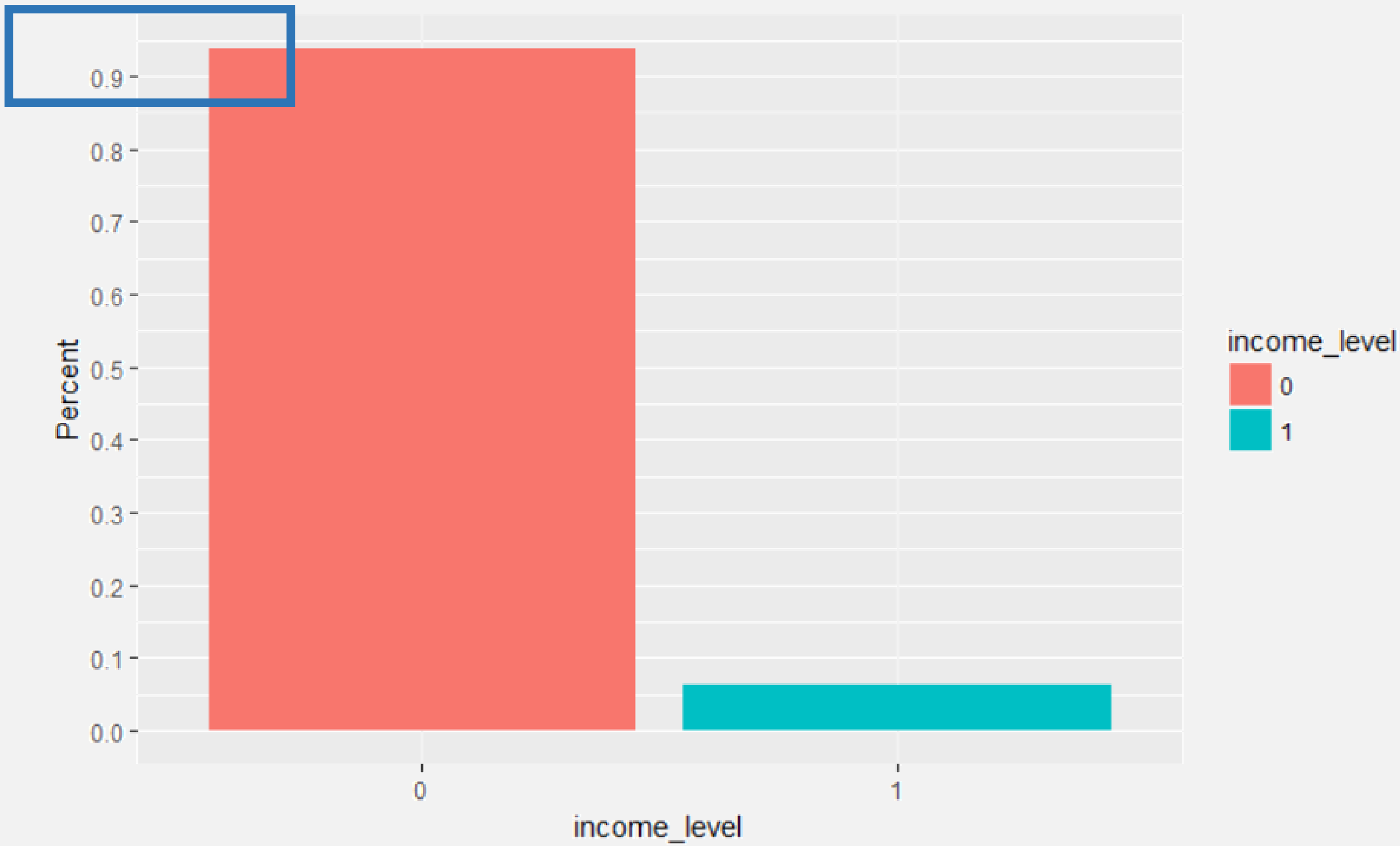
- Larger than $50000: 1
- Less than $50000: 0

# III Resample Data

## Response: income_level

- Larger than $50000: 1
- Less than $50000: 0

About 93.8% of people’ s income less than $50000 a year.



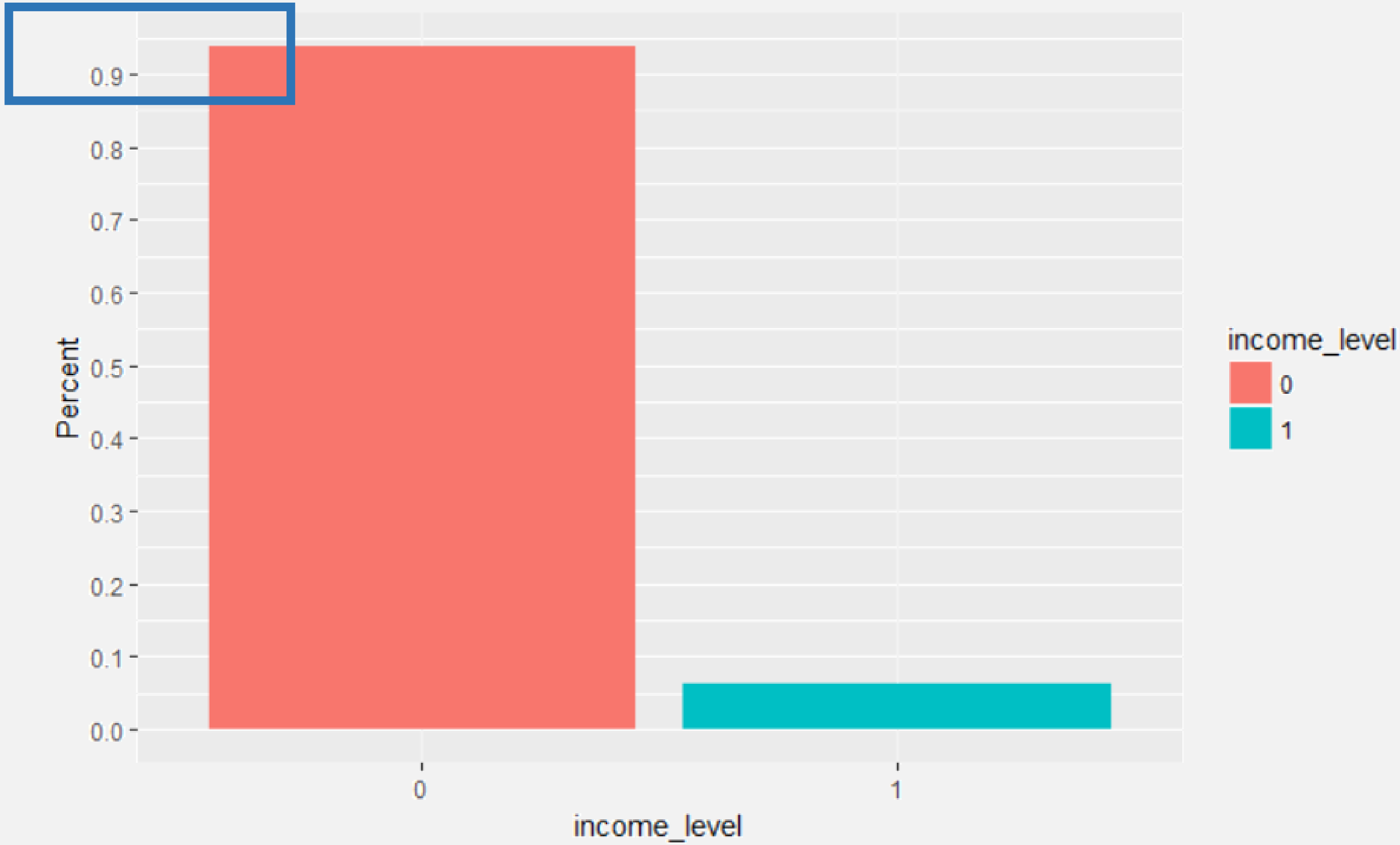
# III Resample Data

Response: income_level

- Larger than $50000: 1
- Less than $50000: 0

About 93.8% of people’ s income less than $50000 a year.

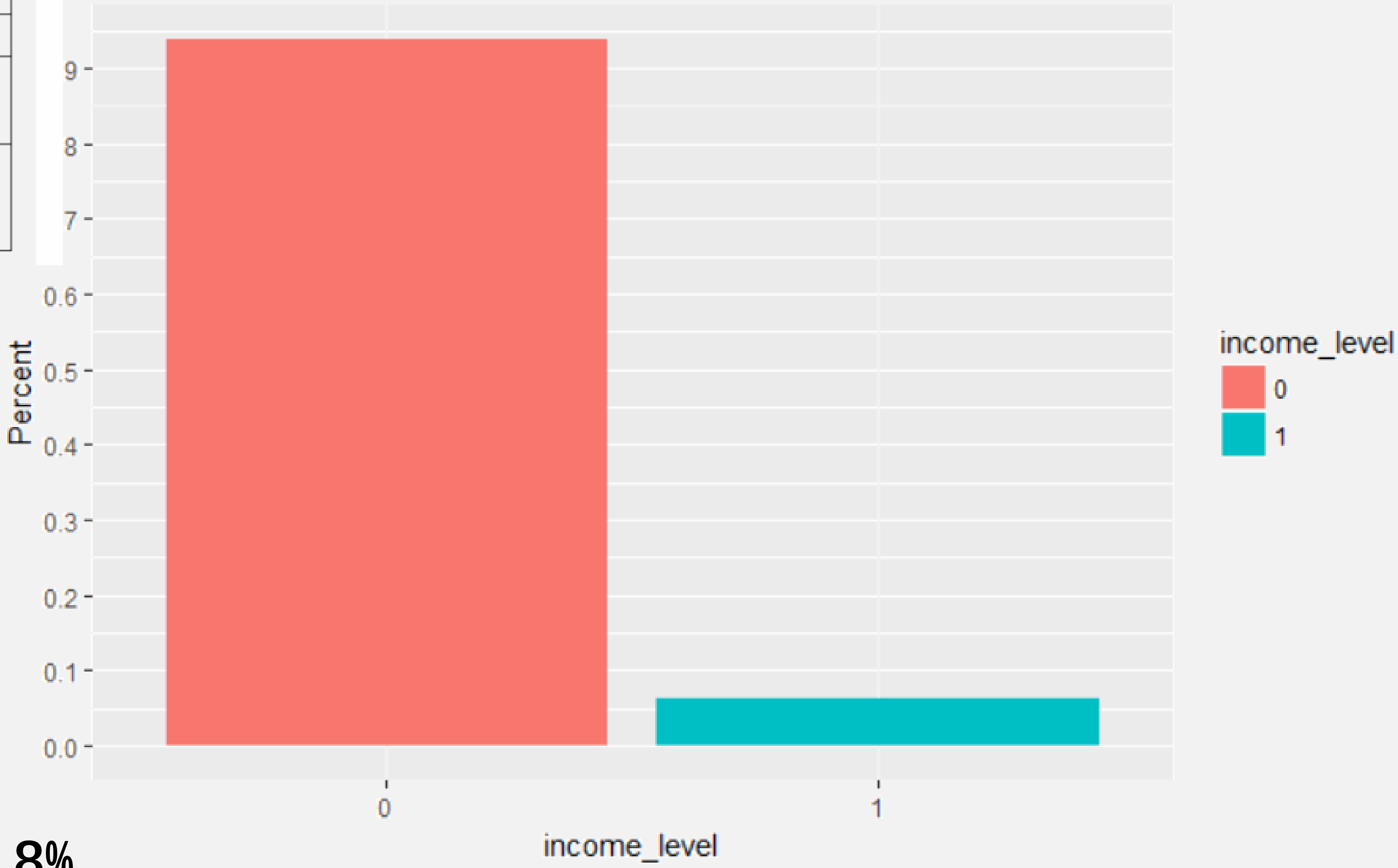
**Imbalanced Dataset**



# III Resample Data

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | True Positive | False Positive |
| | Negative | False Negative | True Negative |

$$\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{P+N} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$$




Model: always returns 0,

Accuracy of prediction will be 93.8%

## III Resample Data

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | True Positive | False Positive |
| | Negative | False Negative | True Negative |

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

# III Resample Data

| | | Actual | |
|---|---|---|---|
| | | Positive | Negative |
| Predicted | Positive | True Positive | False Positive |
| | Negative | False Negative | True Negative |

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

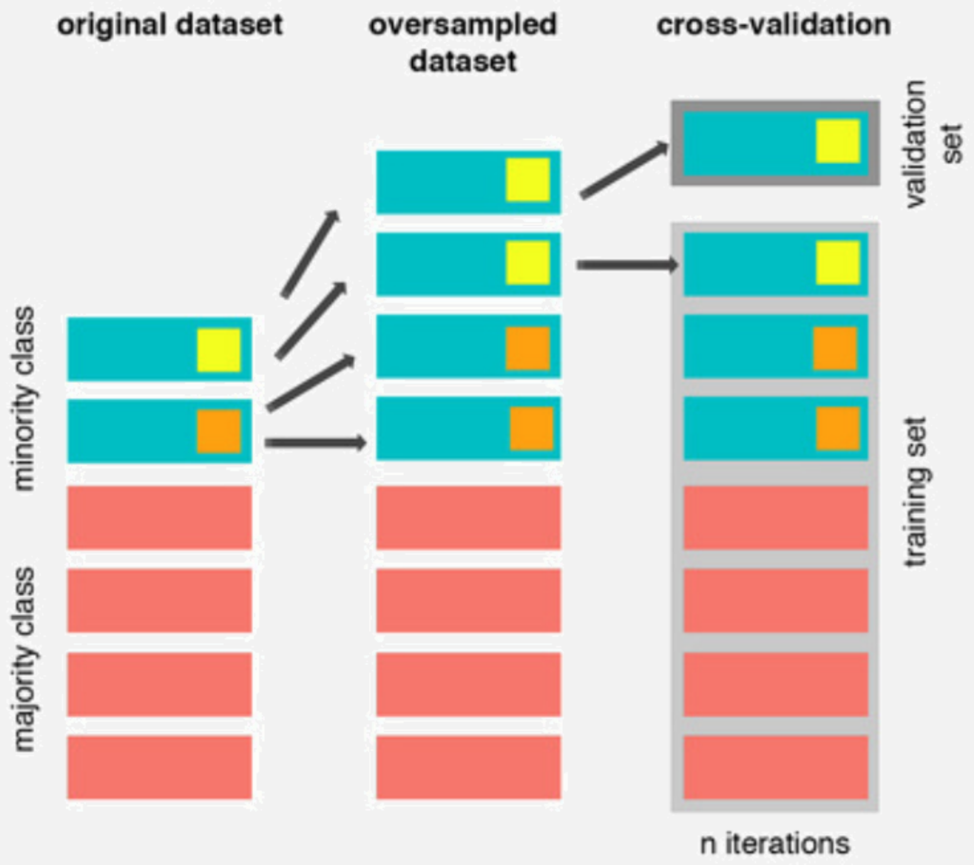
**Original Sample**

```
## precision: 0.726
## recall: 0.157
```

# III Resample Data

Oversample

Undersample



# IV Machine Learning – Logistics Regression

Oversample

```
table(over$income_level)
```

```
     0      1
187141 186698
```

Undersample

```
table(under$income_level)
```

```
    0     1
12211 12382
```

# IV Machine Learning – Logistics Regression

## Oversample

```
table(over$income_level)
```

```
     0      1
187141 186698
```

```
  pred.logit.1
        0     1
0 70372 23204
1  1449  4737
```

## Undersample

```
table(under$income_level)
```

```
    0     1
12211 12382
```

```
  pred.logit.2
        0     1
0 69907 23669
1  1430  4756
```

# IV Machine Learning – Logistics Regression

**Oversample**

```
table(over$income_level)
```

```
     0      1
187141 186698
```

```
  pred.logit.1
       0     1
0  70372 23204
1   1449  4737
```

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \approx 75.20\%$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \approx 97.98\%$$

**Undersample**

```
table(under$income_level)
```

```
    0     1
12211 12382
```

```
  pred.logit.2
       0     1
0  69907 23669
1   1430  4756
```

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \approx 74.71\%$$

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \approx 98\%$$

# IV Machine Learning – Naïve Bayes & Cross Validation

a) Combind dataset (prepared for NaiveBayes & Cross Validation)

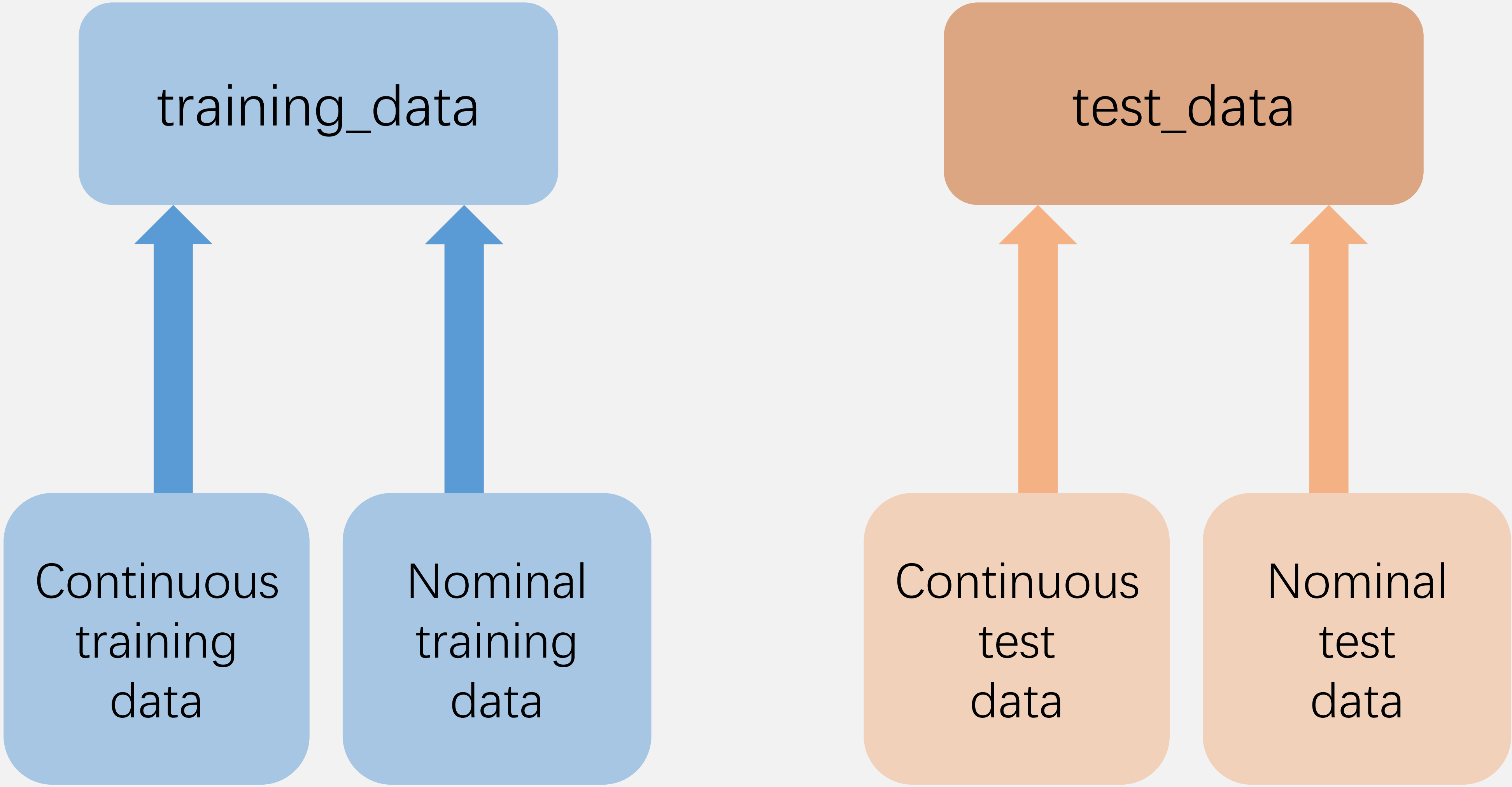
```{r}
train_bind <- cbind(cont_training, nom_training)
test_bind <- cbind(cont_test, nom_test)

train.task <- makeClassifTask(data = train_bind, target = "income_level")
test.task <- makeClassifTask(data = test_bind, target = "income_level")
```

b) Then remove constant data from both datasets.

```{r}
train.task <- removeConstantFeatures(train.task)
test.task <- removeConstantFeatures(test.task)
```

# IV Machine Learning – Naïve Bayes & Cross Validation



# IV Machine Learning – Naïve Bayes & Cross Validation

## Undersample

```
Resampling: cross-validation
Measures:             acc       tp        tn        fp        fn
[Resample] iter 1:    0.8061093   1449.00000001058.0000000181.0000000 422.0000000
[Resample] iter 2:    0.8128015   1469.00000001058.0000000180.0000000 402.0000000
[Resample] iter 3:    0.8054037   1457.00000001047.0000000191.0000000 414.0000000
[Resample] iter 4:    0.8227726   1486.00000001072.0000000166.0000000 385.0000000
[Resample] iter 5:    0.8076552   1449.00000001062.0000000176.0000000 422.0000000
[Resample] iter 6:    0.8022508   1465.00000001030.0000000208.0000000 407.0000000
[Resample] iter 7:    0.8073335   1446.00000001064.0000000174.0000000 425.0000000
[Resample] iter 8:    0.8151125   1479.00000001056.0000000182.0000000 393.0000000
[Resample] iter 9:    0.8064931   1462.00000001047.0000000192.0000000 410.0000000
[Resample] iter 10:   0.8138264   1477.00000001054.0000000184.0000000 395.0000000


Aggregated Result:
acc.test.mean=0.8099759,tp.test.mean=1463.9000000,tn.test.mean=1054.8000000,fp.test.mean=183.4000000,fn.test.mean=407.5000000


acc.test.mean  tp.test.mean  tn.test.mean  fp.test.mean  fn.test.mean
    0.8099759  1463.9000000  1054.8000000   183.4000000   407.5000000
```

# IV Machine Learning – Naïve Bayes & Cross Validation

## Oversample

```
Resampling: cross-validation
Measures:             acc        tp         tn         fp         fn
[Resample] iter 1:    0.8191327    14374.000000016169.00000002404.0000000 4340.0000000
[Resample] iter 2:    0.8161558    14356.000000016076.00000002497.0000000 4358.0000000
[Resample] iter 3:    0.8166117    14426.000000016023.00000002550.0000000 4288.0000000
[Resample] iter 4:    0.8184354    14518.000000015999.00000002574.0000000 4196.0000000
[Resample] iter 5:    0.8194325    14466.000000016089.00000002484.0000000 4249.0000000
[Resample] iter 6:    0.8157267    14381.000000016035.00000002538.0000000 4333.0000000
[Resample] iter 7:    0.8144662    14389.000000015980.00000002593.0000000 4325.0000000
[Resample] iter 8:    0.8218414    14563.000000016081.00000002492.0000000 4151.0000000
[Resample] iter 9:    0.8159144    14470.000000015953.00000002620.0000000 4244.0000000
[Resample] iter 10:   0.8163435    14442.000000015997.00000002576.0000000 4272.0000000


Aggregated Result:
acc.test.mean=0.8174060,tp.test.mean=14438.5000000,tn.test.mean=16040.2000000,fp.test.mean=2532.8000000,fn.test.mean=4275.6000000


acc.test.mean  tp.test.mean  tn.test.mean  fp.test.mean  fn.test.mean
  8.17406e-01   1.44385e+04   1.60402e+04   2.53280e+03   4.27560e+03
```

# IV Machine Learning – Naïve Bayes & Cross Validation

## Undersample

$$Precision\_undersample = \frac{1463.9}{1463.9 + 183.4} \approx 88.87\%$$

$$Recall\_undersample = \frac{1463.9}{1463.9 + 407.5} \approx 78.22\%$$

## Oversample

$$Precision\_oversample = \frac{14438.5}{14438.5 + 2532.8} \approx 85.08\%$$

$$Recall\_oversample = \frac{14438.5}{14438.5 + 4275.6} \approx 77.15\%$$

# IV Machine Learning – Naïve Bayes & Cross Validation

## Undersample

$$Precision\_undersample = \frac{1463.9}{1463.9 + 183.4} \approx 88.87\%$$

$$Recall\_undersample = \frac{1463.9}{1463.9 + 407.5} \approx 78.22\%$$

```
## precision: 0.726
## recall: 0.157
```

## Oversample

$$Precision\_oversample = \frac{14438.5}{14438.5 + 2532.8} \approx 85.08\%$$

$$Recall\_oversample = \frac{14438.5}{14438.5 + 4275.6} \approx 77.15\%$$

# IV Machine Learning – Naïve Bayes & Cross Validation

| acc.test.mean | tp.test.mean | tn.test.mean | fp.test.mean | fn.test.mean |
|---|---|---|---|---|
| 0.8099759 | 1463.9000000 | 1054.8000000 | 183.4000000 | 407.5000000 |

| acc.test.mean | tp.test.mean | tn.test.mean | fp.test.mean | fn.test.mean |
|---|---|---|---|---|
| 8.17406e-01 | 1.44385e+04 | 1.60402e+04 | 2.53280e+03 | 4.27560e+03 |

Whatever Oversample or Undersample, this model can predict the income of people **pretty good**!

# IV Machine Learning – Naïve Bayes & Cross Validation

| acc.test.mean | tp.test.mean | tn.test.mean | fp.test.mean | fn.test.mean |
|---|---|---|---|---|
| 0.8099759 | 1463.9000000 | 1054.8000000 | 183.4000000 | 407.5000000 |

| acc.test.mean | tp.test.mean | tn.test.mean | fp.test.mean | fn.test.mean |
|---|---|---|---|---|
| 8.17406e-01 | 1.44385e+04 | 1.60402e+04 | 2.53280e+03 | 4.27560e+03 |

Whatever Oversample or Undersample, this model can predict the income of people **pretty good**!

Model also works with **nominal** or **continuous** or **combined** datasets.

# V Conclusion

- **Data Separation** is required to remove high correlated variables;

# V Conclusion

- **Data Separation** is required to remove high correlated variables;

- For Imbalanced Data:
    1. **Ignoring** the problem;
    2. **Undersampling** the majority class;
    3. **Oversampling** the minority class:

# V Conclusion

- **Data Separation** is required to remove high correlated variables;

- For Imbalanced Data:
    1. **Ignoring** the problem;
    2. **Undersampling** the majority class;
    3. **Oversampling** the minority class:

- **Boosting** related methods can normally get better prediction to medium size data

# THANK YOU!


- Name： LIN DENG
- Major：Industrial & System Engineering
- Mail: deng0068@umn.edu