

STOR 320 Modeling I

Lecture 15

Yao Li

Department of Statistics and Operations Research

UNC Chapel Hill

Introduction

- Read Vigorously
 - Part IV in R4DS
 - Chapters 6 and 7 in MD
- Goal: Understand the Relationship Between Variables
- Purpose:
 - Explanation
 - Prediction
- Classic Model:
 - Single Outcome Variable (Y)
 - Multiple Predictor Variables (X_1, X_2, \dots, X_P)
 - Multiple Regression Model:
$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_P X_P + \varepsilon$$

Linear Regression

- Model Deconstruction:

$$Y = \underbrace{\beta_0 + \beta_1 X_1 + \cdots + \beta_P X_P}_{\text{Signal}} + \varepsilon$$

↓
Noise

- Noise: Unexplainable Error
 - $E(\varepsilon) = 0$
 - $Var(\varepsilon) = \sigma^2$
- Signal: Helps Us Understand in the Variation in Y
 - $E(Y|X_1, \dots, X_P)$ = Expected Value of Y Given Information about X_1, \dots, X_P
 - Used For Prediction

Linear Regression

- Once We Have Data
 - Estimate the Parameters

$$(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_P)$$

- Use for Prediction

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \dots + \hat{\beta}_P X_P$$

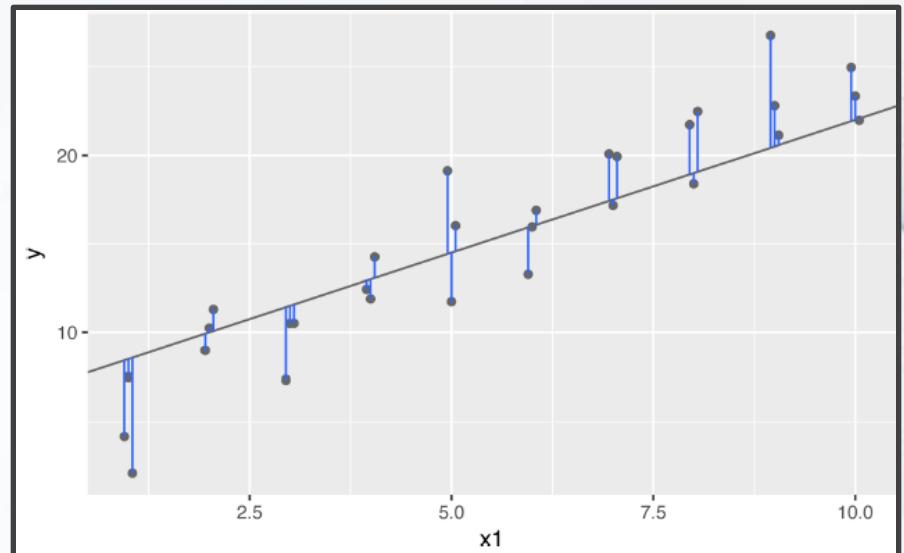
- Obtain the Residuals

$$\hat{\varepsilon} = Y - \hat{Y}$$

- Evaluate the Noise ($\hat{\sigma}^2$)
- Key: Pick Estimates $(\hat{\beta}_0, \hat{\beta}_1, \dots, \hat{\beta}_P)$ where $\hat{\varepsilon} \approx 0$ and $\hat{\sigma}^2$ is Small

Optimization

- Optimization Problem:
 - One Dependent Variable (Y) $y_1, y_2, y_3 \dots, y_n$
 - One Independent Variable (X) $x_1, x_2, x_3 \dots, x_n$
 - Choice of $\hat{\beta}_0$ and $\hat{\beta}_1$
$$\begin{aligned}\hat{\epsilon}_k &= y_k - \hat{y}_k \\ &= y_k - (\hat{\beta}_0 + \hat{\beta}_k x_k)\end{aligned}$$



Optimization

- Optimization Problem (Cont.):
 - Loss Functions:

- Sum of Squared Errors $SSE = \sum \hat{\varepsilon}_k^2$
- Mean Squared Error $MSE = \frac{1}{N} \sum \hat{\varepsilon}_k^2$
- Root MSE $RMSE = \sqrt{\frac{1}{N} \sum \hat{\varepsilon}_k^2}$

- Mean Absolute Error $MAE = \frac{1}{N} \sum |\hat{\varepsilon}_k|$

Family of Models

- Family of Models:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

- Empty Model: $Y = \beta_0 + \varepsilon$
- 1 Coefficient:
 - $Y = \beta_0 + \beta_1 X_1 + \varepsilon$
 - $Y = \beta_0 + \beta_2 X_2 + \varepsilon$
- 2 Coefficients:
 - $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$
- Fact: Adding More Predictor Variables Will Always Cause the Loss Function to Decrease

Data Partition

- Good Practice:
 - Randomly Split Full Dataset Into Two Datasets
 - Training Data
 - 80%-90% of Original Data
 - Used for Model Fitting
 - Testing Data
 - 20%-10% of Original Data
 - Used for Model Selection

Example

- Modeling Real Experimental Data
 - Question: What Factors Improve Hourly Wage?
 - Hypothesis 1: Experience
 - Hypothesis 2: Education



Example

- Modeling Real Experimental Data
 - Data From 10,000 Individuals
 - X_1 = Experience (# of Years)
 - X_2 = Education (# of Years)
 - Y = Salary (dollars/hour)
 - Preview of Data:

```
## # A tibble: 6 x 3
##   salary experience education
##   <dbl>       <int>      <int>
## 1 47.9        27          9
## 2 37.8        24          2
## 3 35.6        19          7
## 4 34.0        17          8
## 5 39.7        25          4
## 6 37.4        23          5
```

Example

```
set.seed(216)
DATA$SPLIT=sample(x=c("TRAIN", "TEST"), size=10000,
                  replace=T, prob=c(0.85, 0.15))
TRAIN=DATA %>% filter(SPLIT=="TRAIN")
TEST=DATA %>% filter(SPLIT=="TEST")
glimpse(TRAIN)
```

```
## Rows: 8,525
## Columns: 4
## $ salary      <dbl> 37.78150, 39.69892, 37.43090, 43.21785, 25.81015, 30.99309...
## $ experience <int> 24, 25, 23, 27, 8, 17, 23, 15, 28, 20, 13, 29, 19, 24, 29, ...
## $ education   <int> 2, 4, 5, 4, 5, 5, 3, 7, 0, 3, 4, 7, 6, 7, 4, 3, 6, 7, 5, 7...
## $ SPLIT       <chr> "TRAIN", "TRAIN", "TRAIN", "TRAIN", "TRAIN", "TRAIN", "TRA...
```

```
glimpse(TEST)
```

```
## Rows: 1,475
## Columns: 4
## $ salary      <dbl> 47.88340, 35.60634, 34.00961, 39.45598, 38.36103, 50.13343...
## $ experience <int> 27, 19, 17, 22, 21, 30, 21, 24, 22, 16, 13, 20, 31, 21, 23...
## $ education   <int> 9, 7, 8, 8, 8, 9, 15, 8, 9, 8, 11, 9, 7, 10, 13, 8, 10, 8, ...
## $ SPLIT       <chr> "TEST", "TEST", "TEST", "TEST", "TEST", "TEST", "TEST", "T...
```

Empty Model (MODEL 0)

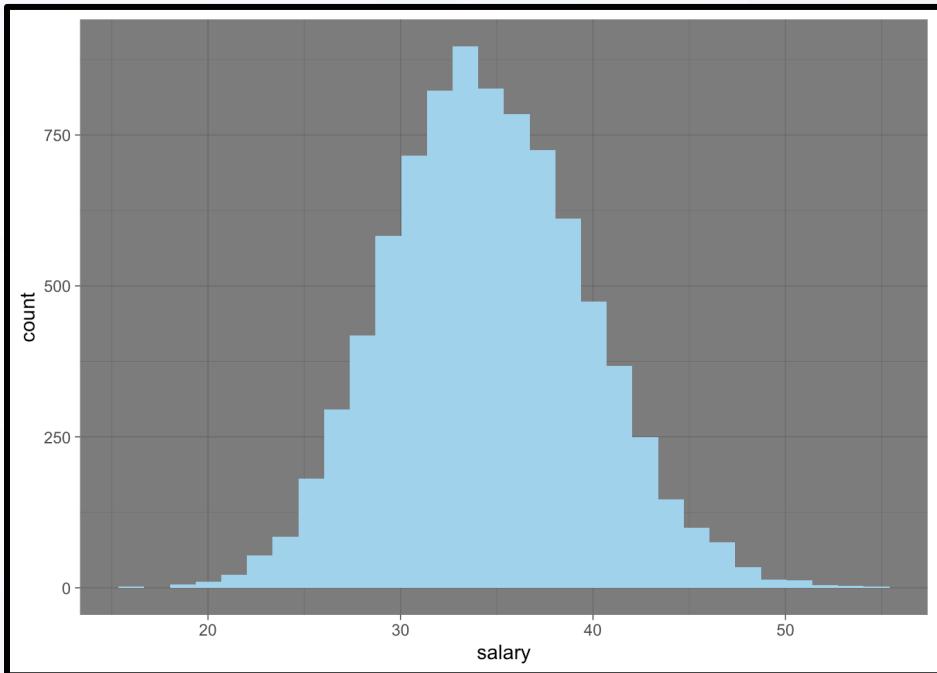
- MODEL 0

$$Y = \beta_0 + \varepsilon$$

$$E(Y) = \beta_0$$

- Summary of Salary

```
## # A tibble: 1 x 4
##       mean     sd    min    max
##   <dbl> <dbl> <dbl> <dbl>
## 1 34.5  5.15 16.1  54.8
```



Empty Model (MODEL 0)

- Function to Get Fitted Values:

```
MODEL0 = function(DATA,COEF){  
  FIT=COEF[1]  
}
```

- Functions to Evaluate Model:

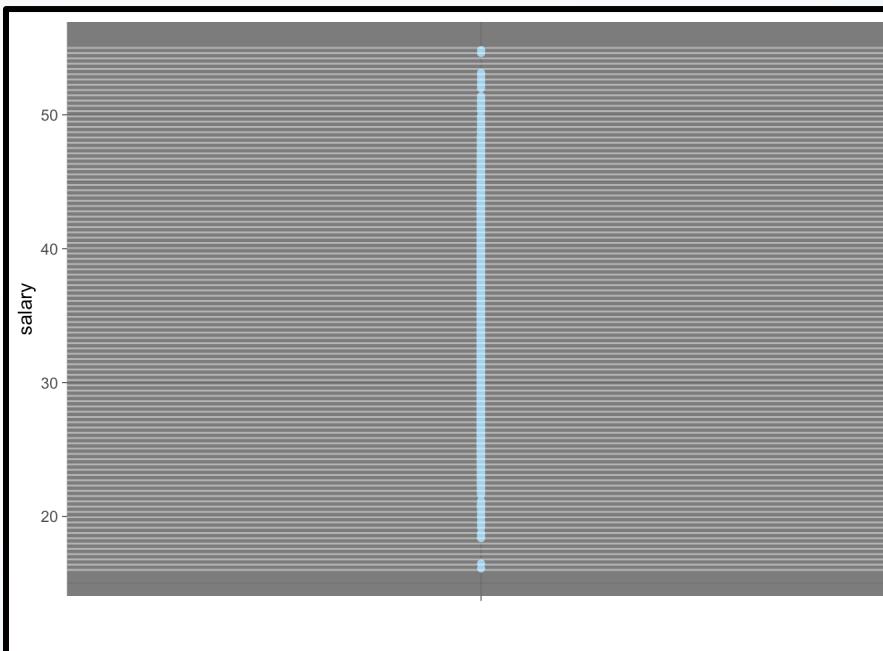
```
MSE0=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL0(DATA,COEF)  
  LOSS=mean(ERROR^2)  
  return(LOSS)  
}  
MAE0=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL0(DATA,COEF)  
  LOSS=mean(abs(ERROR))  
  return(LOSS)  
}
```

Empty Model: Optimization

- Optimization
 - Specify Possible Values of $\hat{\beta}_0$

```
COEF0=tibble(  
  beta0=seq(16,55,length=100)  
)
```

- All Possible Models



Empty Model: Optimization

- Optimization
 - We Desire to Find the $\hat{\beta}_0$ that Minimizes MSE and MAE
 - map(): purr Package

```
COEF0 %>%
  mutate(MSE=purrr::map_dbl(beta0,MSE0,DATA=TRAIN),
         MAE=purrr::map_dbl(beta0,MAE0,DATA=TRAIN),
         rankMSE=rank(MSE),rankMAE=rank(MAE)) %>%
  filter(rankMSE<5,rankMAE<5)
```

```
## # A tibble: 3 x 5
##   beta0     MSE     MAE rankMSE rankMAE
##   <dbl>   <dbl>   <dbl>    <dbl>    <dbl>
## 1 34.1    26.7   4.13      3        2
## 2 34.5    26.5   4.13      1        1
## 3 34.9    26.7   4.15      2        3
```

Empty Model: Optimization

- Optimization
 - optim(): Base R

```
BESTMSE0=optim(par=16,fn=MSE0,DATA=TRAIN)  
BESTMSE0$par
```

```
## [1] 34.53125
```

Starting Values

```
BESTMAE0=optim(par=16,fn=MAE0,DATA=TRAIN)  
BESTMAE0$par
```

```
## [1] 34.34375
```

- lm(): Base R (Linear Reg)

```
LM0=lm(salary~1,data=TRAIN)  
coef(LM0)
```

```
## (Intercept)  
## 34.53428
```

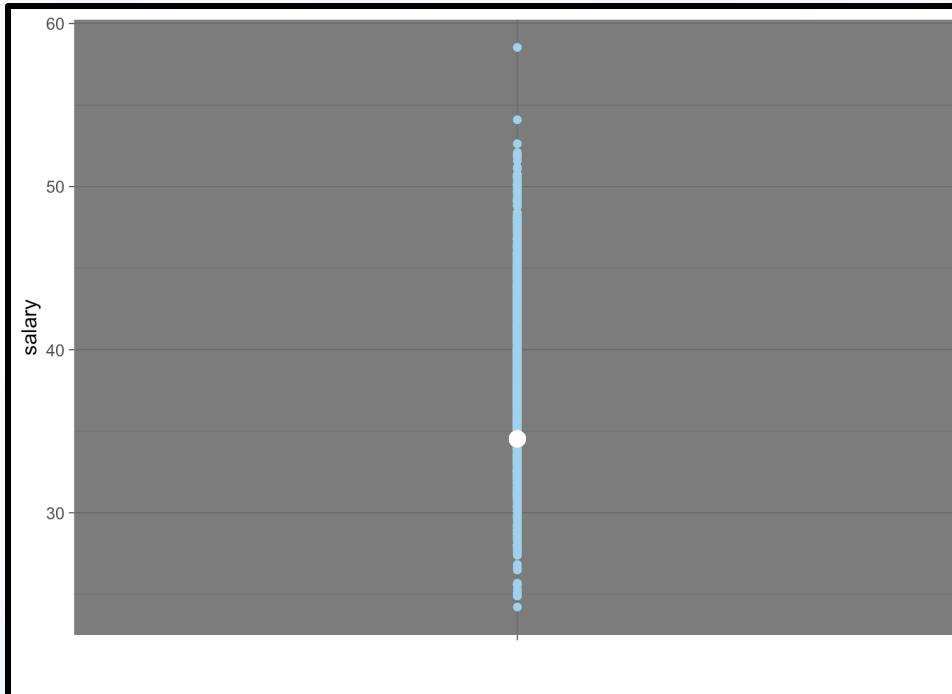
Empty Model: Final Model

- Final MODEL 0

$$Y = 34.53 + \varepsilon$$

$$E(Y) = 34.53$$

- Prediction on Test Data:

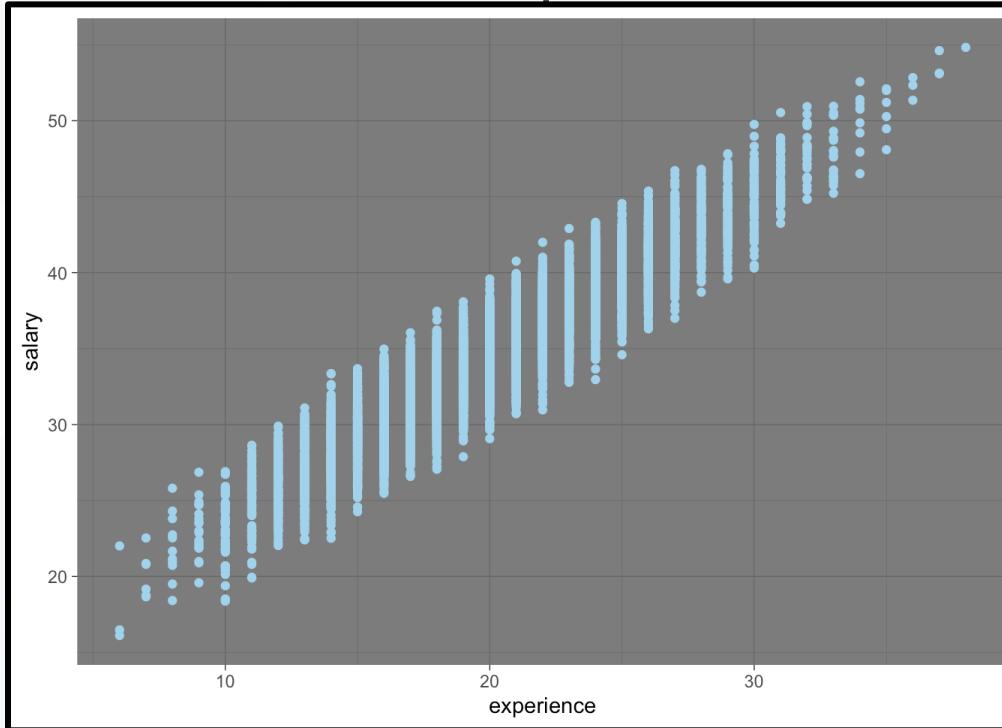


Model 1A

- MODEL 1A

$$Y = \beta_0 + \beta_1 X_1 + \varepsilon$$
$$E(Y) = \beta_0 + \beta_1 X_1$$

- Visualization of Relationship



Model 1A

- Function to Get Fitted Values

```
MODEL1A = function(DATA,COEF){  
  FIT=COEF[1]+COEF[2]*DATA$experience  
}
```

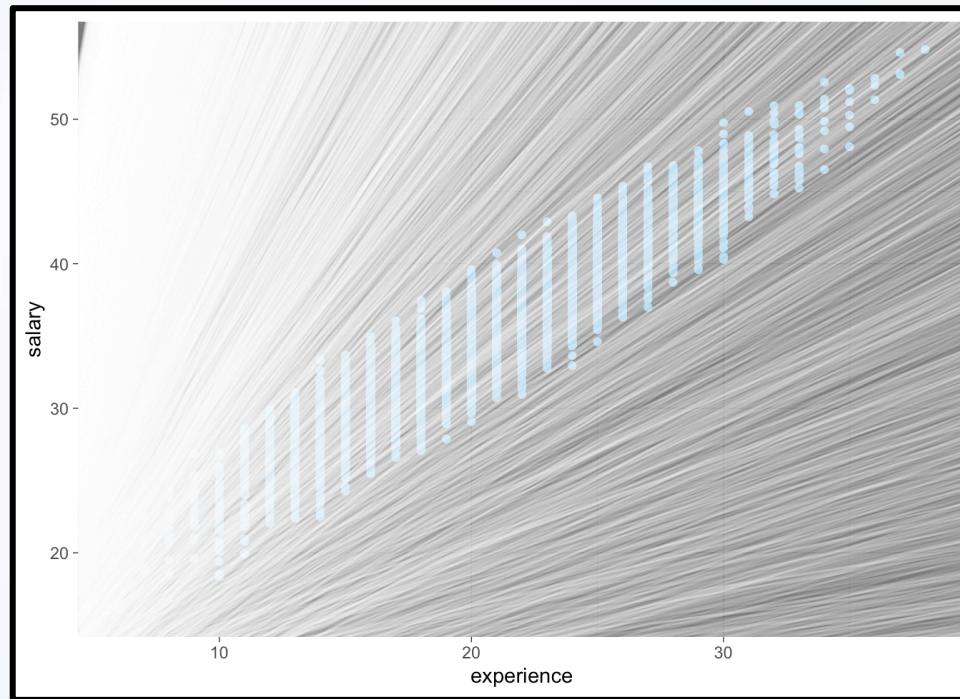
- Functions to Evaluate Model

```
MSE1A=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL1A(DATA,COEF)  
  LOSS=mean(ERROR^2)  
  return(LOSS)  
}  
  
MAE1A=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL1A(DATA,COEF)  
  LOSS=mean(abs(ERROR))  
  return(LOSS)  
}
```

Model 1A: Optimization

- Optimization
 - Possible Values of $\hat{\beta}_0$ and $\hat{\beta}_1$
 - All Possible Models

```
set.seed(216)
COEF1A=tibble(
  beta0=runif(10000,0,10),
  beta1=runif(10000,0,10)
)
```



Model 1A: Optimization

- Optimization
 - Use of apply() Function

```
COEF1A %>%
  mutate(MSE=apply(COEF1A, 1, MSE1A, DATA=TRAIN),
        MAE=apply(COEF1A, 1, MAE1A, DATA=TRAIN),
        rankMSE=rank(MSE), rankMAE=rank(MAE)) %>%
  filter(rankMSE<5, rankMAE<5)
```

```
## # A tibble: 4 x 6
##   beta0    beta1     MSE     MAE  rankMSE  rankMAE
##   <dbl>   <dbl>   <dbl>   <dbl>     <dbl>     <dbl>
## 1  9.48   1.26   4.09   1.63      3         3
## 2  9.46   1.23   4.00   1.62      2         2
## 3  9.84   1.20   4.12   1.65      4         4
## 4  9.40   1.24   3.96   1.61      1         1
```

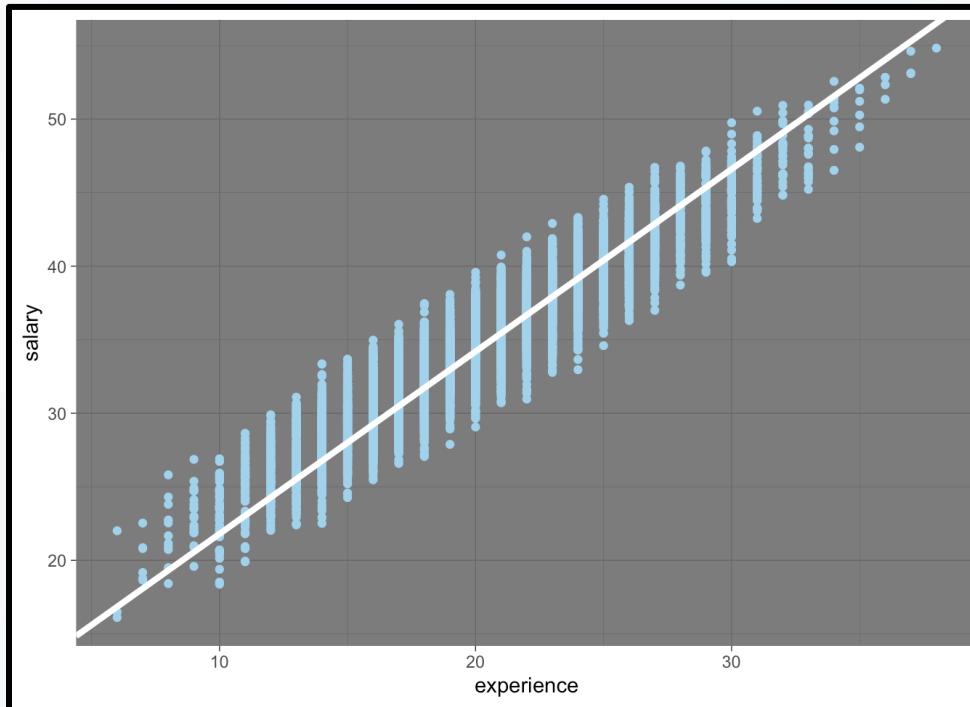
Model 1A: Final Model

- Final MODEL 1A

$$Y = 9.4 + 1.24X_1 + \varepsilon$$

$$E(Y) = 9.4 + 1.24X_1$$

- Fitted on Train Data



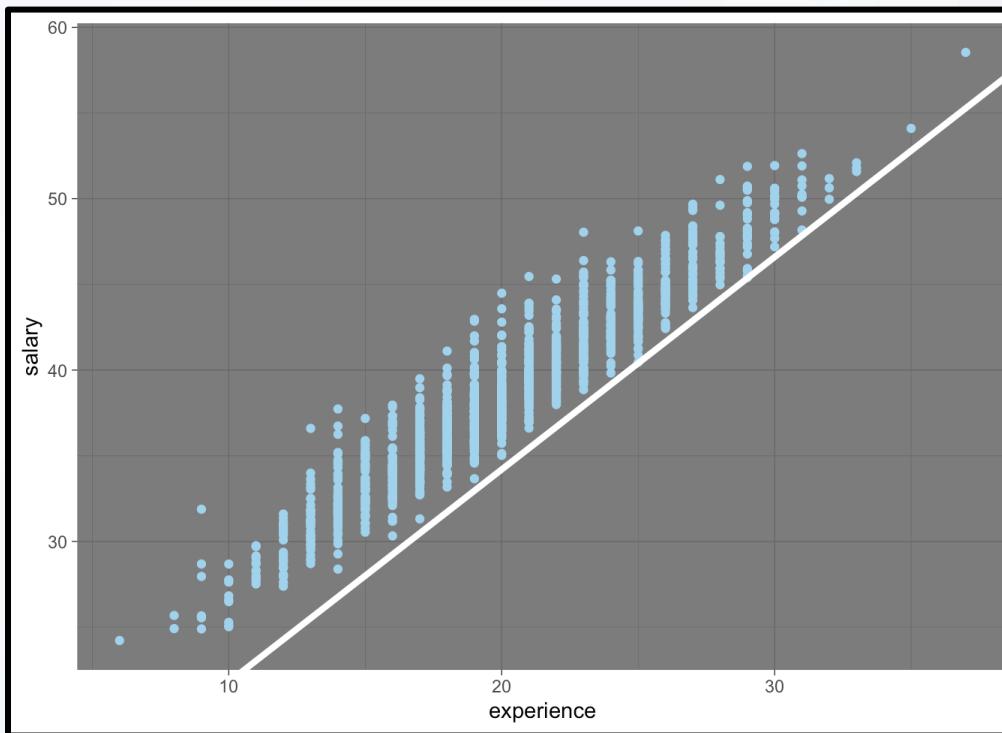
Model 1A: Final Model

- Final MODEL 1A

$$Y = 9.4 + 1.24X_1 + \varepsilon$$

$$E(Y) = 9.4 + 1.24X_1$$

- Prediction on Test Data



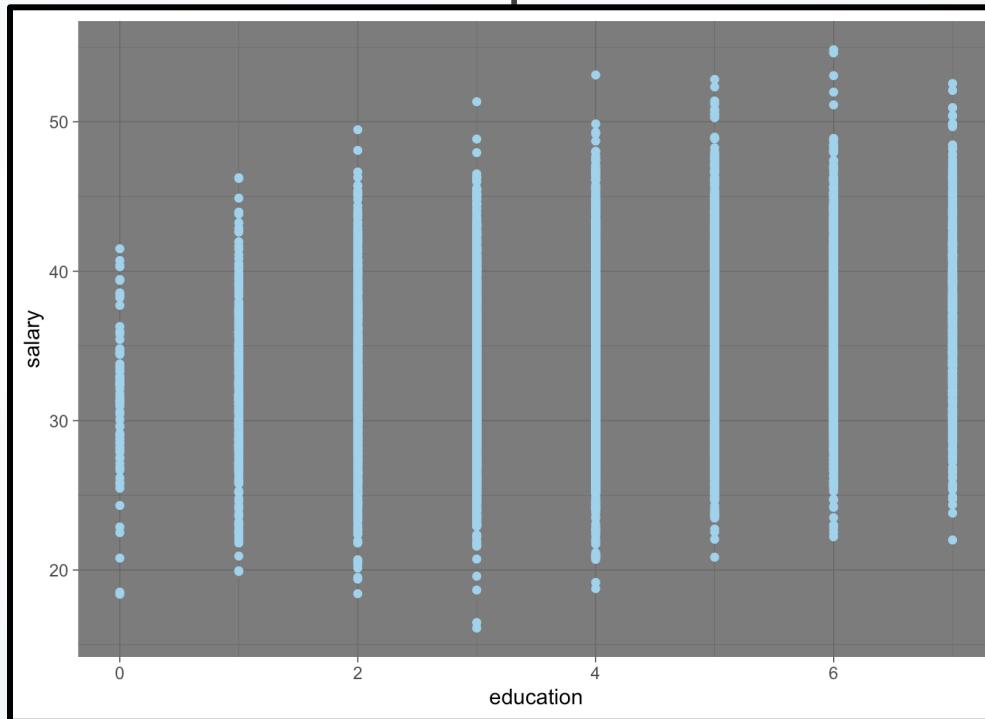
Model 1B

- MODEL 1B

$$Y = \beta_0 + \beta_1 X_2 + \varepsilon$$

$$E(Y) = \beta_0 + \beta_1 X_2$$

- Visualization of Relationship



Model 1B

- Function to Get Fitted Values

```
MODEL1B = function(DATA,COEF){  
  FIT=COEF[1]+COEF[2]*DATA$education  
}
```

- Functions to Evaluate Model

```
MSE1B=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL1B(DATA,COEF)  
  LOSS=mean(ERROR^2)  
  return(LOSS)  
}  
  
MAE1B=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL1B(DATA,COEF)  
  LOSS=mean(abs(ERROR))  
  return(LOSS)  
}
```

Model 1B: Optimization

- Optimization
 - Use of optim() Function

```
BESTMSE1B=optim(par=c(0,0),fn=MSE1B,DA  
TA=TRAIN)  
BESTMSE1B$par
```

```
## [1] 30.8323639 0.8543225
```

```
BESTMAE1B=optim(par=c(0,0),fn=MAE1B,DA  
TA=TRAIN)  
BESTMAE1B$par
```

```
## [1] 30.6619753 0.8512186
```

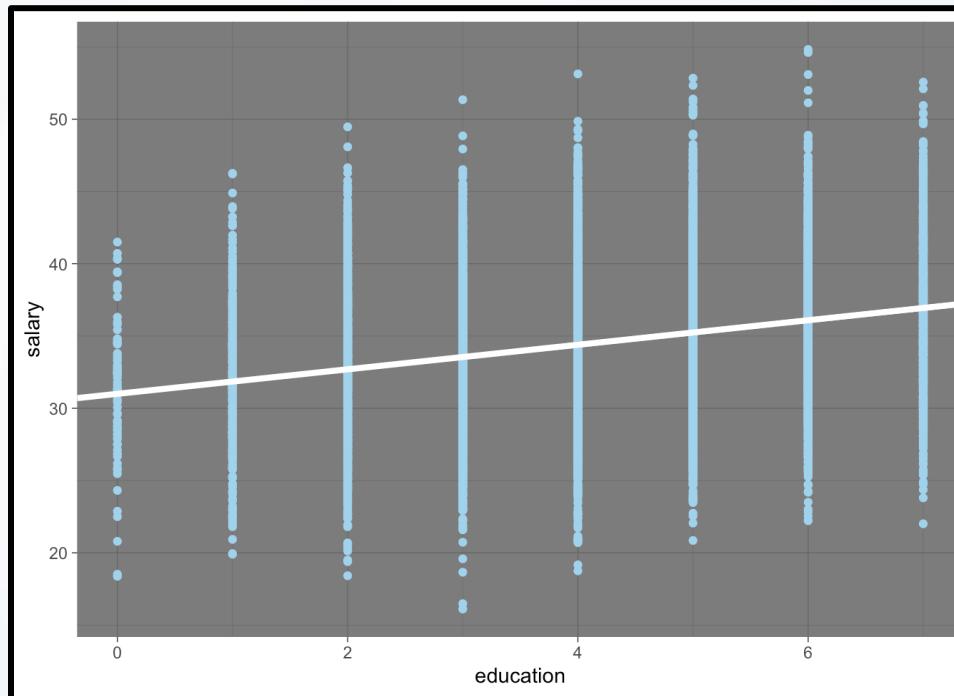
Model 1B: Final Model

- Final MODEL 1B

$$Y = 31 + 0.85X_2 + \varepsilon$$

$$E(Y) = 31 + 0.85X_2$$

- Fitted on Train Data



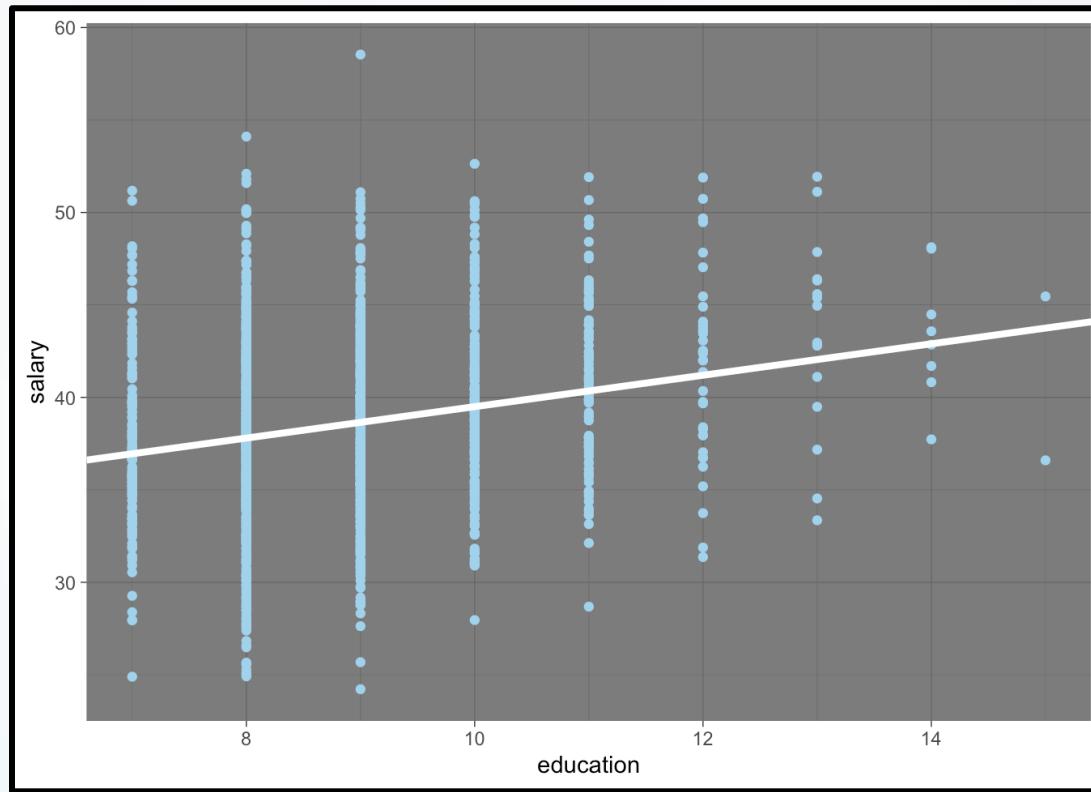
Model 1B: Final Model

- Final MODEL 1B

$$Y = 31 + 0.85X_2 + \varepsilon$$

$$E(Y) = 31 + 0.85X_2$$

- Prediction on Test Data



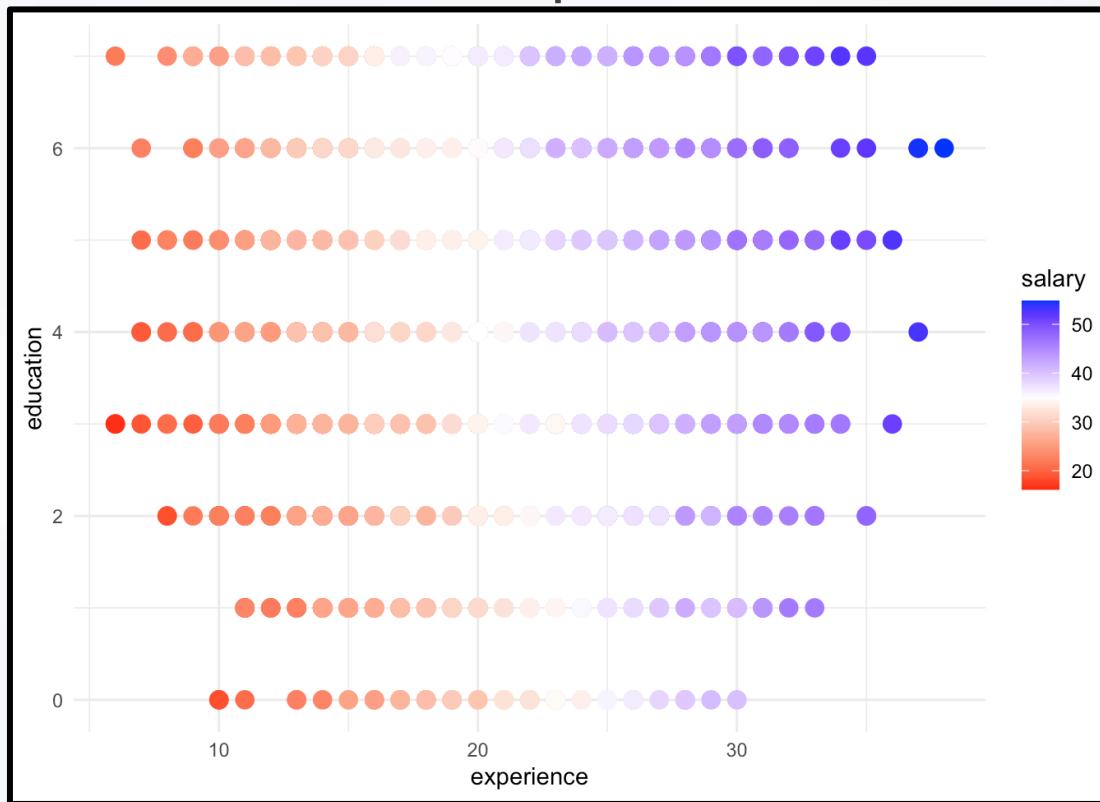
MODEL 2

- MODEL 2

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \varepsilon$$

$$E(Y) = \beta_0 + \beta_1 X_1 + \beta_2 X_2$$

- Visualization of Relationship



MODEL 2

- Function to Get Fitted Values

```
MODEL2 = function(DATA,COEF){  
  FIT=COEF[1]+COEF[2]*DATA$experience+COEF[3]*DATA$education  
}
```

- Functions to Evaluate Model

```
MSE2=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL2(DATA,COEF)  
  LOSS=mean(ERROR^2)  
  return(LOSS)  
}  
MAE2=function(DATA,COEF){  
  ERROR=DATA$salary-MODEL2(DATA,COEF)  
  LOSS=mean(abs(ERROR))  
  return(LOSS)  
}
```

Multiple Regression

- Use lm() with summary()
- Final MODEL 2

$$Y = 9 + 1.08X_1 + 0.9X_2 + \varepsilon$$

$$E(Y) = 9 + 1.08X_1 + 0.9X_2$$

```
LM2=lm(salary~experience+education,data=TRAIN)
summary(LM2)

##
## Call:
## lm(formula = salary ~ experience + education, data = TRAIN)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6426 -0.6776 -0.0138  0.6838  3.7675
##
## Coefficients:
##             Estimate Std. Error t value     Pr(>|t|)
## (Intercept) 8.996672  0.058760 153.1 <0.0000000000000002 ***
## experience  1.079243  0.002474 436.3 <0.0000000000000002 ***
## education    0.902851  0.006635 136.1 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.025 on 8522 degrees of freedom
## Multiple R-squared:  0.9605, Adjusted R-squared:  0.9604
## F-statistic: 1.035e+05 on 2 and 8522 DF,  p-value: < 0.0000000000000022
```

Model Summary

```
LM2=lm(salary~experience+education,data=TRAIN)
```

```
summary(LM2)
```

```
##
```

```
## Call:
```

```
## lm(formula = salary ~ experience + education, data = TRAIN)
```

```
##
```

```
## Residuals:
```

```
##      Min      1Q  Median      3Q     Max
```

```
## -3.6426 -0.6776 -0.0138  0.6838  3.7675
```

```
##
```

```
## Coefficients:
```

```
##             Estimate Std. Error t value          Pr(>|t|)
```

```
## (Intercept) 8.996672   0.058760 153.1 <0.0000000000000002 ***
```

```
## experience  1.079243   0.002474 436.3 <0.0000000000000002 ***
```

```
## education    0.902851   0.006635 136.1 <0.0000000000000002 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

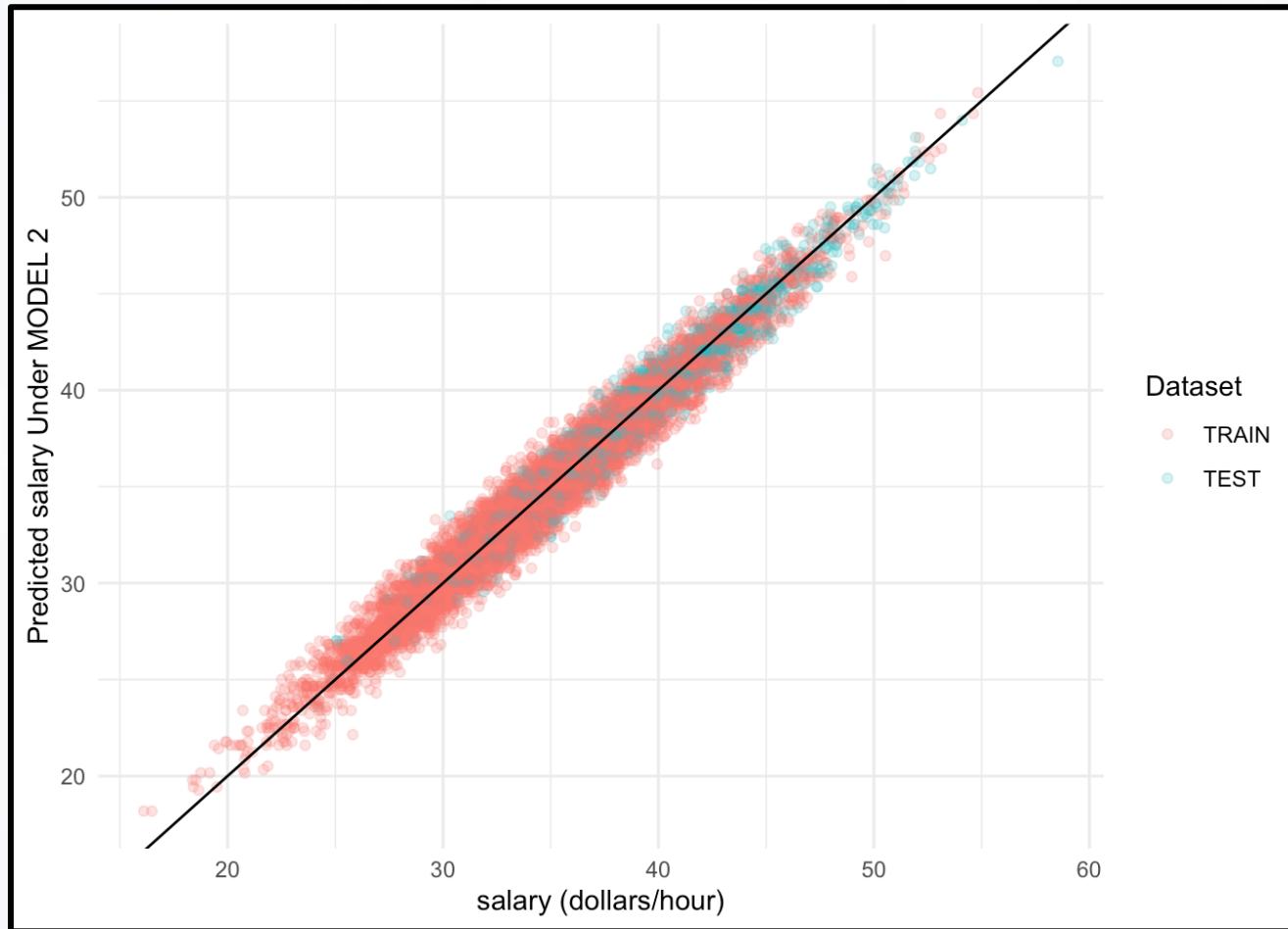
```
## Residual standard error: 1.025 on 8522 degrees of freedom
```

```
## Multiple R-squared:  0.9605, Adjusted R-squared:  0.9604
```

```
## F-statistic: 1.035e+05 on 2 and 8522 DF,  p-value: < 0.0000000000000022
```

Visualization

- Comparing Predicted Values to Actual Values for MODEL 2



Model Evaluation

- Out-of-Sample Evaluation

```
MODELS=c("MODEL 0", "MODEL 1A", "MODEL 1B", "MODEL 2")
MSE=c(MSE0(TEST,c(34.53)),
      MSE1A(TEST,c(9.4,1.24)),
      MSE1B(TEST,c(31,0.85)),
      MSE2(TEST,c(9,1.07,0.9)))
MAE=c(MAE0(TEST,c(34.53)),
      MAE1A(TEST,c(9.4,1.24)),
      MAE1B(TEST,c(31,0.85)),
      MAE2(TEST,c(9,1.07,0.9)))
COMPARE=tibble(MODELS=MODELS,MSE=MSE,MAE=MAE)
print(COMPARE)
```

```
## # A tibble: 4 x 3
##   MODELS     MSE     MAE
##   <chr>     <dbl>  <dbl>
## 1 MODEL 0    42.0    5.17
## 2 MODEL 1A   21.5    4.31
## 3 MODEL 1B   24.5    3.94
## 4 MODEL 2    0.965  0.786
```