Train - Test Sample Approach

Contents

Read Data **1**

Split Data **1**

Train Sample **3**

Estimate model 3

Visualize Estimated Model 4

Evaluate Model 4

Test Sample **5**

Apply fitted model to test sample 5

Visualize Estimated Model on test sample 5

Evaluate Regression Model using test sample 6

Evaluating a model using data to estimate it risks overfitting the model to the data used to build it. For this reason, it is best to evaluate the model using a different dataset. Since getting a fresh dataset for evaluating a model may be costly, time consuming and in some cases impossible, it is common to split the data into a train and test sample, estimating the model using the train and evaluating it using the test.

The purpose of this note is to illustrate this approach of using the train sample to estimate the model and the test sample to evaluate it using a data on wages. Specifically, we are interested in predicting `earn` using `height`.

Read Data

```
wages = read.csv('wages.csv')
str(wages)
```

```
## 'data.frame':    1379 obs. of  6 variables:
##  $ earn  : int  159142 192794 97422 160956 164178 30626 94208 101920 6426 85994 ...
##  $ height: num  73.9 66.2 63.8 63.2 63.1 ...
##  $ gender: Factor w/ 2 levels "female","male": 2 1 1 1 1 1 1 2 2 2 ...
##  $ race  : Factor w/ 4 levels "african-american",..: 4 4 4 2 4 4 4 4 3 4 ...
##  $ ed    : int  16 16 16 16 17 15 12 17 15 12 ...
##  $ age   : int  49 62 33 95 43 30 53 50 25 30 ...
```

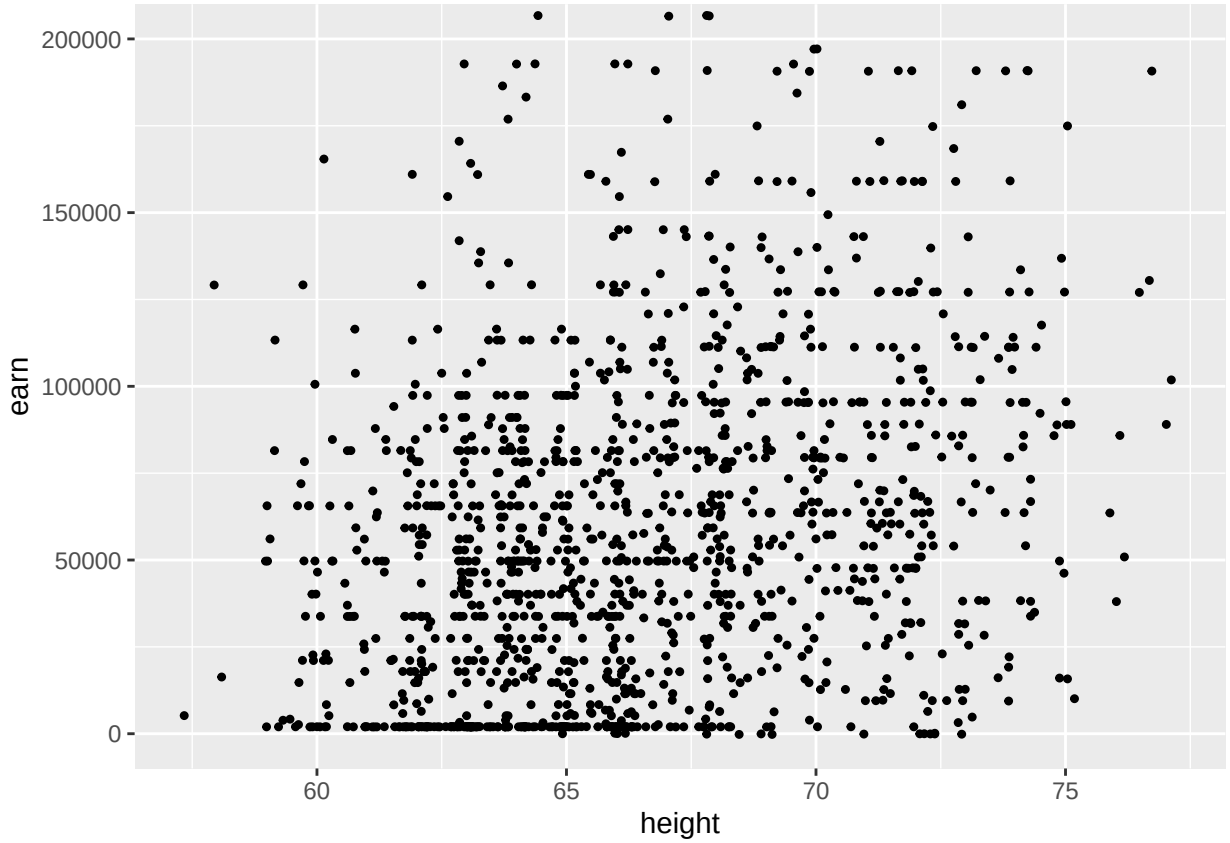
Split Data

Split data in a 70:30 ratio such that 70% of the observations go to the train sample and the remaining 30% to the test sample.

```
set.seed(103111)
split = sample(1:nrow(wages),0.7*nrow(wages))
train = wages[split,]
test = wages[-split,]
```

Now, let us visualize the train and test samples using a scatterplot as a lens. Since we are interested in examining the relationship between **earn** and **height**, lets construct a scatterplot to represent these variables.

```
library(ggplot2)
ggplot(data=wages,aes(x=height,y=earn))+
  geom_point(size=0.9)+
  coord_cartesian(ylim=c(0,200000))
```

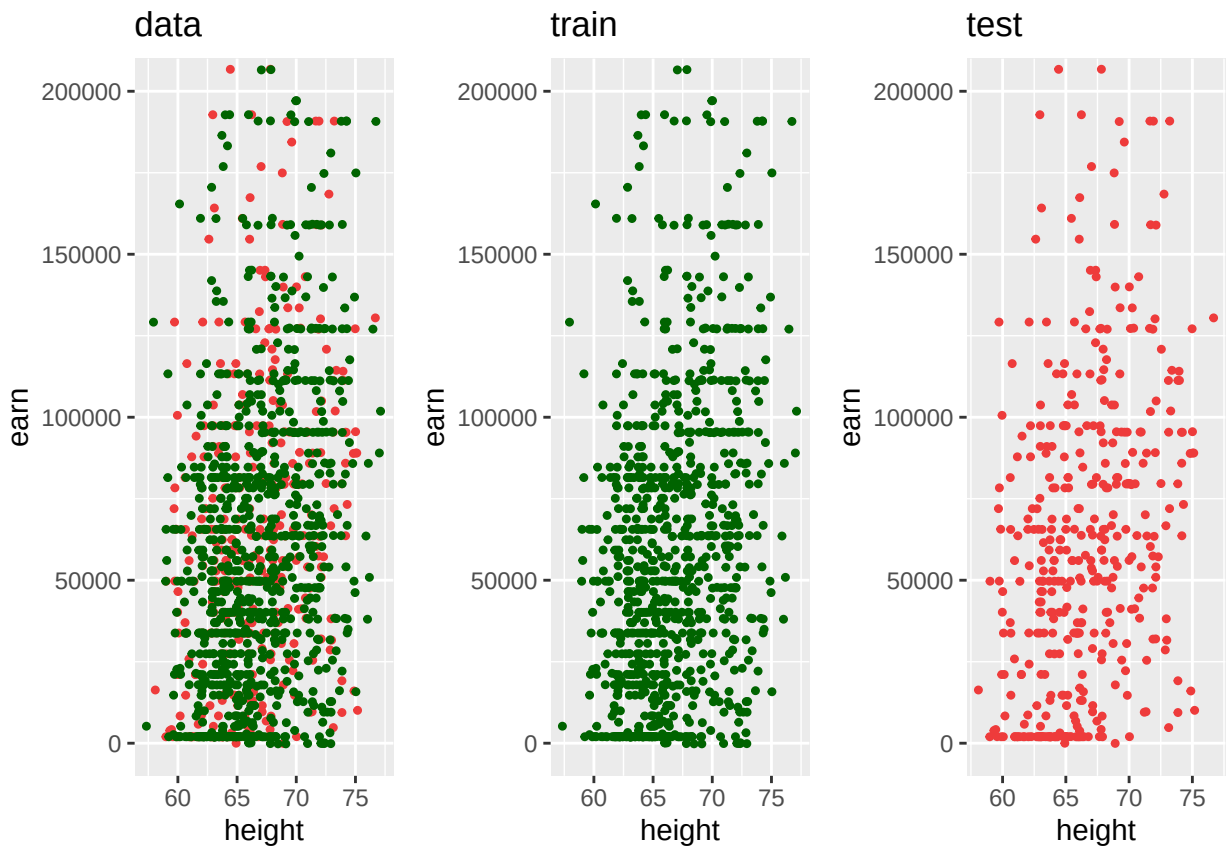


Next, let us distinguish points in the scatterplot based on whether it is in the train or test sample

```
g_data =
  ggplot()+
  geom_point(data=test,aes(x=height,y=earn), color='brown2',size=0.9)+
  geom_point(data=train,aes(x=height,y=earn), color='darkgreen',size=0.9)+
  coord_cartesian(ylim=c(0,200000))+
  ggtitle('data')
g_train =
  ggplot()+
  geom_point(data=train,aes(x=height,y=earn), color='darkgreen',size=0.9)+
  coord_cartesian(ylim=c(0,200000))+ggtitle('train')
```

```
g_test =
  ggplot()+
  geom_point(data=test,aes(x=height,y=earn), color='brown2',size=0.9)+
  coord_cartesian(ylim=c(0,200000))+ggtitle('test')

library(gridExtra)
grid.arrange(g_data, g_train,g_test,nrow=1)
```



Visually speaking, our goal is to fit a line to the green dots and then evaluate it with respect to the red dots.

Train Sample

Estimate model

Model is estimated using the train sample

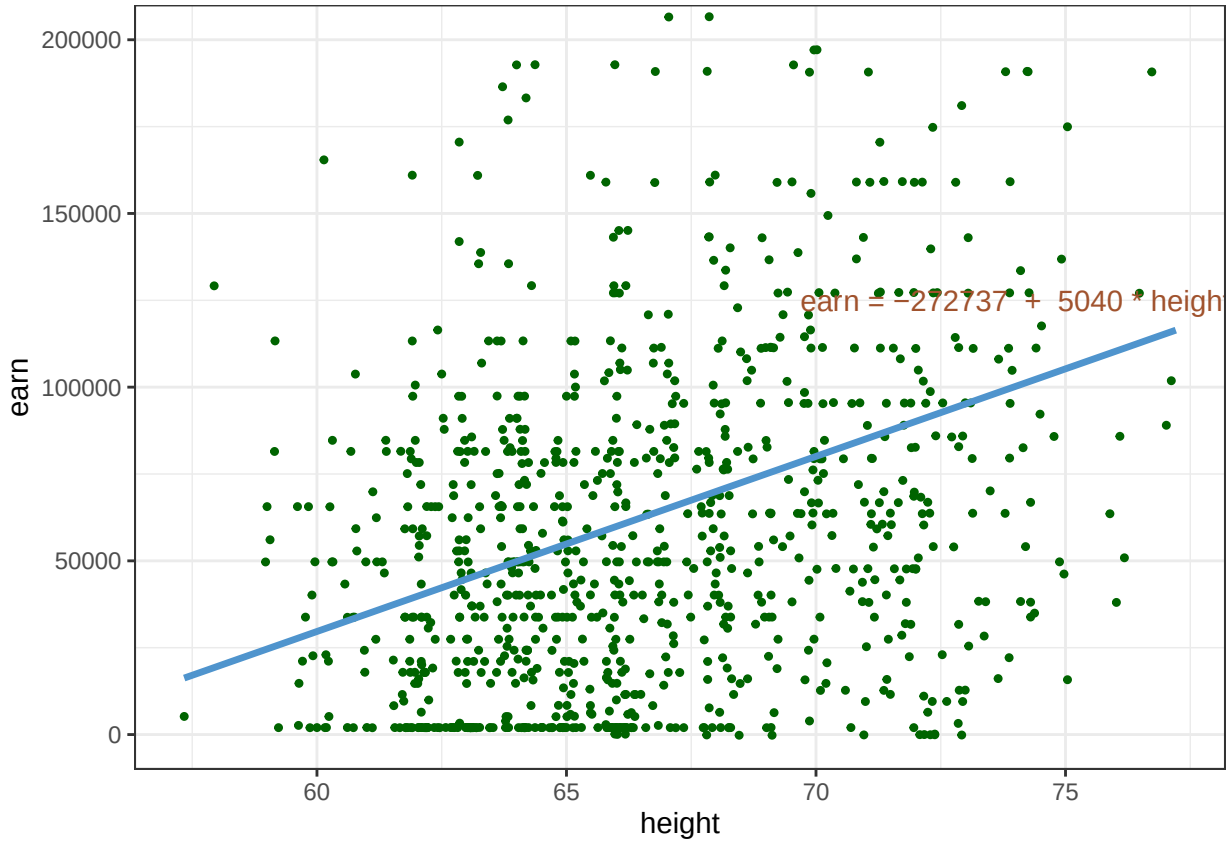
```
model = lm(earn~height,train)
equation = noquote(paste('earn =',round(model$coefficients[1],0),' + ',round(model$coefficients[2],0),'
equation
```

```
## [1] earn = -272737  +  5040 * height
```

Visualize Estimated Model

The fitted line is such that the (sum of squared) distance from the green dots (train sample) is minimum.

```
ggplot(data=train,aes(x=height,y=earn))+
  geom_point(color='darkgreen',size=0.9)+
  geom_smooth(method="lm",size=1.2,color="steelblue3",se=FALSE)+
  coord_cartesian(ylim=c(0,200000))+
  theme_bw()+
  annotate('text',label=equation,x=74,y=125000,color='sienna')
```



Evaluate Model

We can compute the root mean squared error to evaluate performance of the model. This is the error of the estimated model based on the data used to build it, also known as in-sample error. This error will in most cases be less than the error from fitting the model to a new dataset.

```
pred_train = predict(model)
rmse_train = sqrt(mean((pred_train - train$earn)^2))
rmse_train
```

```
## [1] 59005.08
```

Test Sample

Apply fitted model to test sample

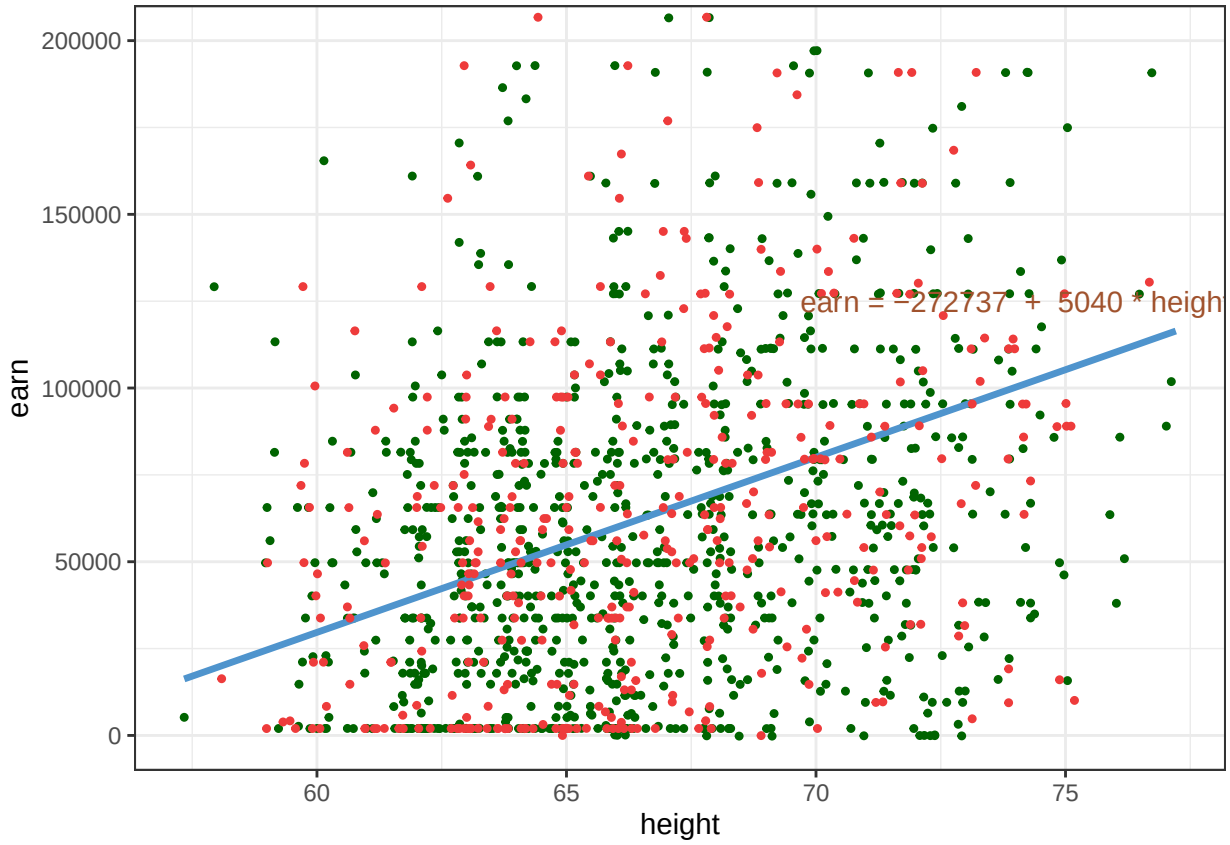
Note, we are not estimating a new regression model, rather simply applying the model developed above to the test sample.

```
pred_test = predict(model,newdata=test)
```

Visualize Estimated Model on test sample

The regression equation is the same as in the earlier plot.

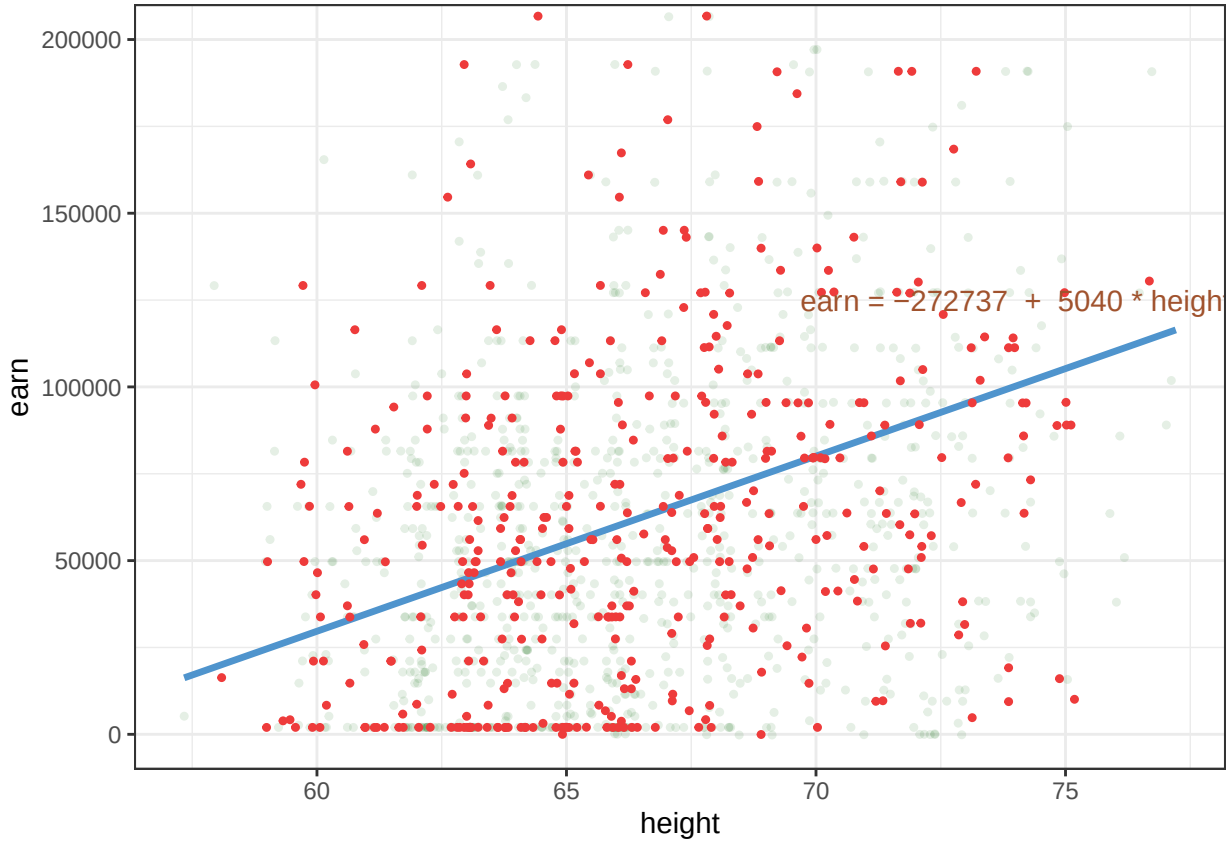
```
ggplot(data=train,aes(x=height,y=earn))+
  geom_point(color='darkgreen',size=0.9)+
  geom_smooth(method="lm",size=1.2,color="steelblue3",se=FALSE)+
    geom_point(data=test,aes(x=height,y=earn),color='brown2',size=0.9)+
  coord_cartesian(ylim=c(0,200000))+
  theme_bw()+
  annotate('text',label=equation,x=74,y=125000,color='sienna')
```



Evaluate Regression Model using test sample

To assess error in the test sample, we compare the regression line to the test data (red dots) not train data (green dots)

```
ggplot(data=train,aes(x=height,y=earn))+
  geom_point(color='darkgreen',size=0.9,alpha=0.1)+
  geom_smooth(method="lm",size=1.2,color="steelblue3",se=FALSE)+
    geom_point(data=test,aes(x=height,y=earn),color='brown2',size=0.9)+
  coord_cartesian(ylim=c(0,200000))+
  theme_bw()+
  annotate('text',label=equation,x=74,y=125000,color='sienna')
```




```
pred_test = predict(model,newdata=test)
rmse_test = sqrt(mean((pred_test - test$earn)^2))
rmse_test
```

```
## [1] 61670.1
```

In closing, the blue line below is derived from the green dots (train sample) but evaluated using the red dots (test sample).