Class 4 Lab

John Kubale

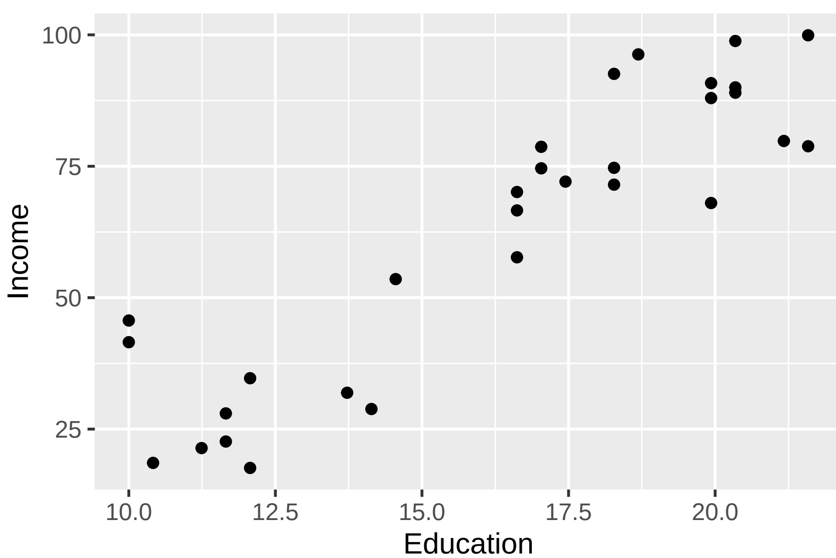
2024-09-17

Use Income2.csv data and `Income` as the response variable

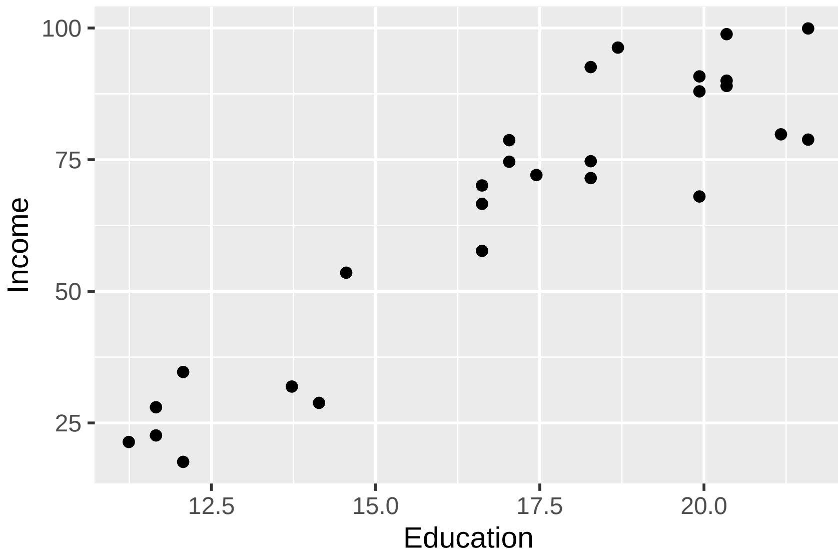
```
inc <- read_csv("~/UMD/classes/stat_mod_ML_1_SURV615/class_3/Income2.csv")
```

1. Make a scatter plot (no regression) of `Income` as a function of `Education`. Remake the scatterplot, this time only including those where $Education > 11$

```
inc |>
  ggplot(aes(x=Education, y=Income)) +
  geom_point()
```




```
inc |>
  filter(Education > 11) |>
  ggplot(aes(x=Education, y=Income)) +
  geom_point()
```



1a. Fit a simple linear regression model with `Income` as a function of `Education`. Interpret the coefficients.

- For everyone 1 year increase in education, income goes up by 6.4, and for a person with no Education, we would expect an average mean income of -41.9.

```
mod_income <- lm(Income ~ Education, inc)
summary(mod_income)
```

```
##
## Call:
## lm(formula = Income ~ Education, data = inc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.568  -8.012   1.474   5.754  23.701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -41.9166     9.7689  -4.291 0.000192 ***
## Education     6.3872     0.5812  10.990 1.15e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.93 on 28 degrees of freedom
## Multiple R-squared:  0.8118, Adjusted R-squared:  0.8051
## F-statistic: 120.8 on 1 and 28 DF,  p-value: 1.151e-11
```

2. Use the code below to manually calculate sXY and sXX based on the formulas we saw in today's lecture.

- From Lecture 4 slides 20-24

$$\hat{\beta}_1 = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \equiv \frac{SS_{XY}(n-1)^{-1}}{SS_X(n-1)^{-1}} = \frac{s_{XY}}{s_{XX}}$$

- where

$$s_{XY} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

-and

$$s_{XX} = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$$

```
SS_XY<-sum((inc$Education-mean(inc$Education))*(inc$Income-mean(inc$Income)))
SS_X<-sum((inc$Education-mean(inc$Education))^2)

SS_XY/(dim(inc)[1]-1)
```

```
## [1] 92.74695
```

```
SS_X/(dim(inc)[1]-1)
```

```
## [1] 14.52084
```

```
s_XY<-cov(inc$Education,inc$Income)
s_XX<-var(inc$Education)
s_XY
```

```
## [1] 92.74695
```

```
s_XX
```

```
## [1] 14.52084
```

3. Use sXY and sXX to calculate $\hat{\beta}_1$ and then calcuate $\hat{\beta}_0$ based on the formulas above.

```
"Beta_1"
```

```
## [1] "Beta_1"
```

```
SS_XY/SS_X
```

```
## [1] 6.387161
```

```
"Beta_0"
```

```
## [1] "Beta_0"
```

```
mean(inc$Income) - (mean(inc$Education) *  SS_XY/SS_X)
```

```
## [1] -41.91661
```

4. Compute $\hat{Y}$ based on the model saving as a new variable in inc called h_Income_edu. Compute $\hat{Y}$ using the predict() function saving as a new variable in inc called h_Income_edu2. Look at the estimates for each variable. How do they compare?

```
beta0 <- coef(mod_income)[[1]]
beta1 <- coef(mod_income)[[2]]

inc$h_Income_edu <- beta0+beta1*inc$Education
inc$h_Income_edu2 <- predict(mod_income)
```

4a. Use the predict() function to get the mean expected income for a person with 10 years of education (Education == 10). Hint: look at the newdata argument in ?predict.lm.

```
new_dat <- data.frame(Education=10)
predict(mod_income, newdata = new_dat)
```

```
##      1
## 21.955
```

4b. Using the code below, plot a histogram of the observed data points overlayed with the fitted/predicted values based on the model. What did the melt() function do? How does the plot look to you?

```
# install.packages("reshape") ## uncomment and run this code if reshape package not yet installed then
library(reshape2)
inc_sub<-melt(inc%>%select(Income,h_Income_edu))
head(inc_sub)
```

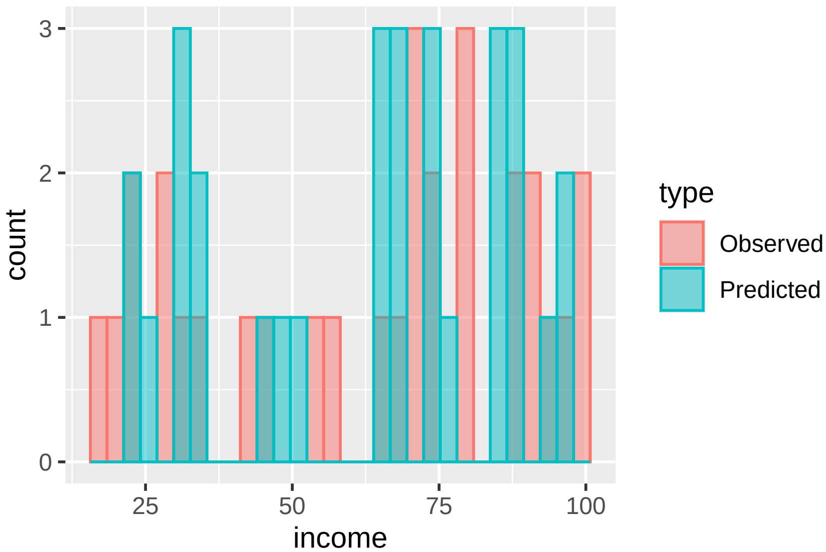
```
##   variable    value
## 1   Income 99.91717
## 2   Income 92.57913
## 3   Income 34.67873
## 4   Income 78.70281
## 5   Income 68.00992
## 6   Income 71.50449
```

```
tail(inc_sub)
```

```
##        variable    value
## 55 h_Income_edu 95.95797
## 56 h_Income_edu 29.88389
## 57 h_Income_edu 85.38612
## 58 h_Income_edu 32.52685
## 59 h_Income_edu 35.16982
## 60 h_Income_edu 66.88538
```

```
inc_sub<-inc_sub%>%
  mutate(type=ifelse(variable=="Income","Observed","Predicted"),
         income=value)

ggplot(inc_sub, aes(x=income, color=type, fill=type)) +
   geom_histogram(position="identity", alpha=0.5)
```



5. How are the standard error of regression coefficients estimated?

- From lecture 4 slides 18-21,

$$\hat{V}(\hat{\beta}_1) = \frac{\hat{\sigma}^2}{SS_X}$$

- , where

$$\hat{\sigma}^2$$

- is estimated error variance (or residual variance) as

$$\hat{\sigma}^2 = \frac{\sum \hat{\epsilon}_i^2}{n-2}$$

$$\hat{V}(\hat{\beta}_0) = \hat{\sigma}^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{SS_X} \right)$$

```
inc$residual_edu<-inc$Income-inc$h_Income_edu
resid(mod_income)
```

```
##           1           2           3           4           5           6
##   3.9592013  17.7648696  -0.4910891  11.8174308 -17.3761966  -3.3097798
##           7           8           9          10          11          12
##   2.5843487 -13.5039782   1.9772456  23.7005295 -13.8278614  18.8257683
##          13          14          15          16          17          18
##  -4.5443481   2.3593803  19.5769925   0.9716193 -19.5683318  -6.5607179
##          19          20          21          22          23          24
##   5.8626839  10.8049300  -0.1095660   2.5045098   2.5505850  -6.0272982
##          25          26          27          28          29          30
## -17.1521870  -8.4953284   5.4279169  -9.8906904 -17.5562232   7.7255848
```

```
cor(inc$residual_edu, resid(mod_income))
```

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

```
h_sigma_sq_edu<-sum(inc$residual_edu^2)/(dim(inc)[1]-2)
h_sigma_sq_edu
```

```
## [1] 142.2324
```

```r
summary(mod_income)
```

```
##
## Call:
## lm(formula = Income ~ Education, data = inc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.568  -8.012   1.474   5.754  23.701
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -41.9166     9.7689  -4.291 0.000192 ***
## Education     6.3872     0.5812  10.990 1.15e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.93 on 28 degrees of freedom
## Multiple R-squared:  0.8118, Adjusted R-squared:  0.8051
## F-statistic: 120.8 on 1 and 28 DF,  p-value: 1.151e-11
```

```r
summary(mod_income)$sigma^2
```

```
## [1] 142.2324
```

```r
SS_X<-sum((inc$Education-mean(inc$Education))^2)
V_beta1<-h_sigma_sq_edu/SS_X
SE_beta1<-sqrt(V_beta1)
V_beta0<-h_sigma_sq_edu*(1/dim(inc)[1]+mean(inc$Education)^2/SS_X)
SE_beta0<-sqrt(V_beta0)
SE_beta0;SE_beta1
```

```
## [1] 9.768949
```

```
## [1] 0.5811716
```

6. Extract the standard errors for model1's parameters from its model summary and compare to the values calculated manually above.

```r
summary(mod_income)$coefficients[1, 2]
```

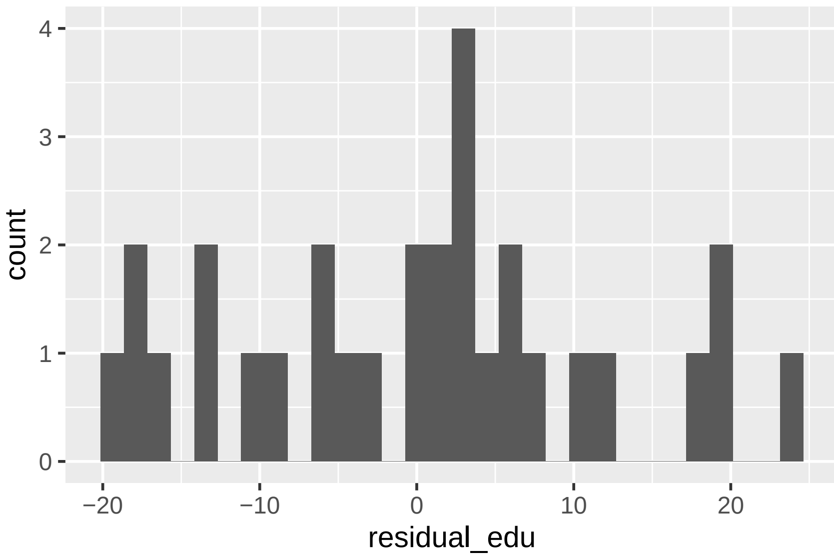
```
## [1] 9.768949
```

```r
summary(mod_income)$coefficients[2, 2]
```

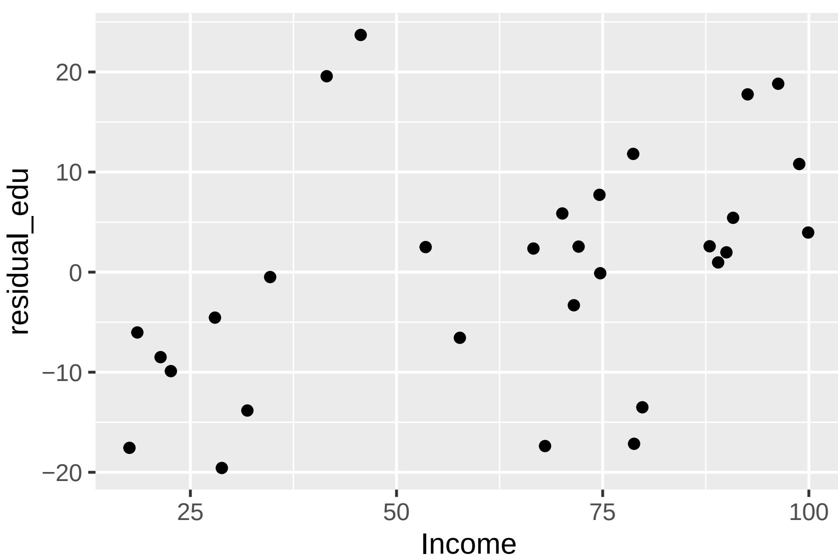
```
## [1] 0.5811716
```

7. Look at the distribution of the residuals from model1 using a histogram. Create a scatter plot of the residuals (y-axis) vs. income (x-axis).

```r
inc |>
  ggplot(aes(x=residual_edu)) +
  geom_histogram()
```



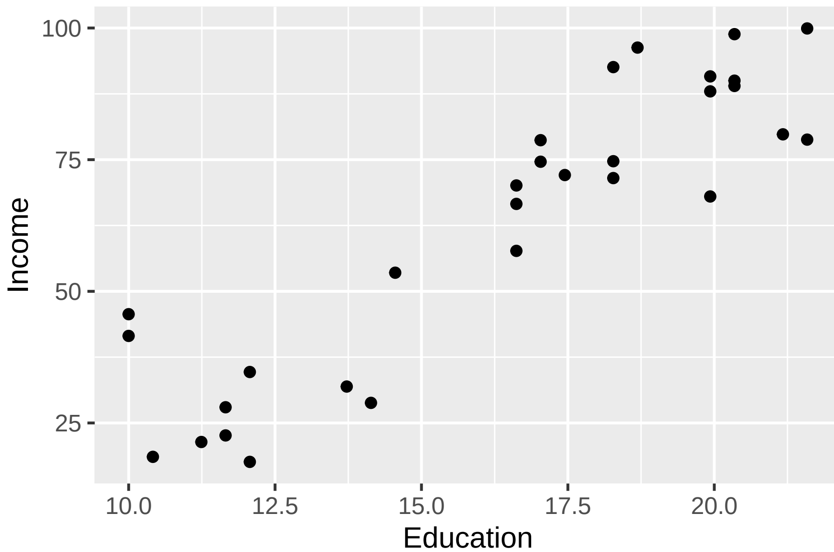

```
inc |>
  ggplot(aes(x=Income, y=residual_edu)) +
  geom_point()
```



Example code for customizing plots in ggplot2.—-

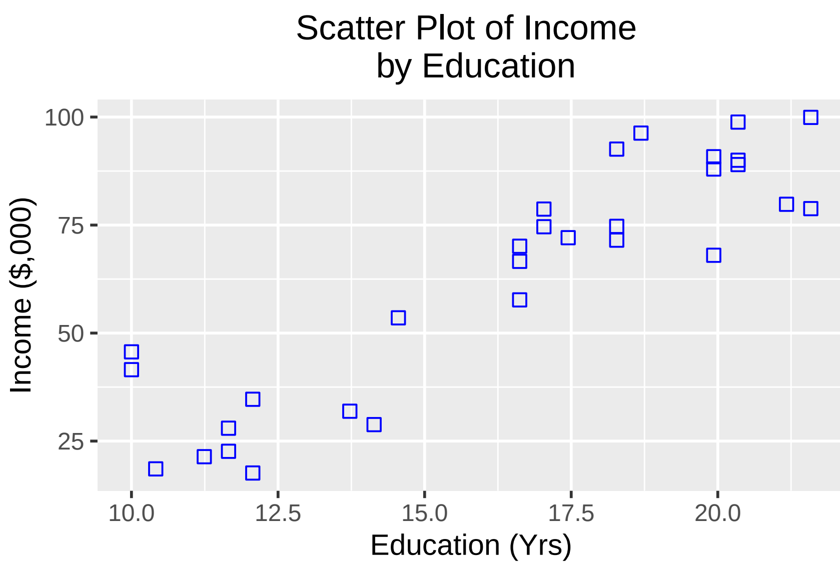
- `Income` as a function of `Education`; no regression implied

```
library(ggplot2)
ggplot(inc, aes(y=Income, x=Education))+
  geom_point()
```



- With aesthetic options added

```
ggplot(inc, aes(y=Income, x=Education))+
  geom_point(shape=0, color="blue", size=2)+
  labs(title="Scatter Plot of Income \n by Education",
        y ="Income ($,000)", x = "Education (Yrs)")+
  theme(plot.title = element_text(hjust = 0.5))
```



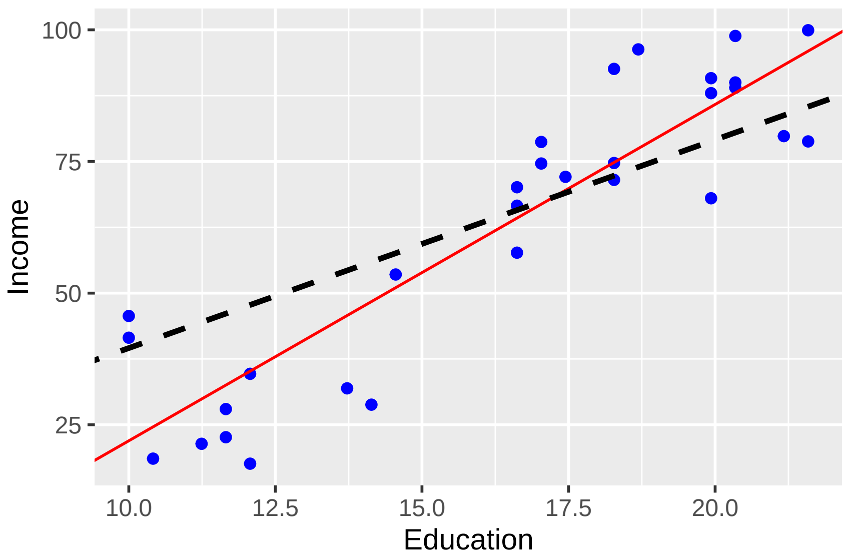

```
m_edu<-lm(Income~Education,inc)
coef(m_edu)
```

```
## (Intercept)   Education
##  -41.916612    6.387161
```

```
m_edu_noint<-lm(Income~Education-1,inc)
coef(m_edu_noint)
```

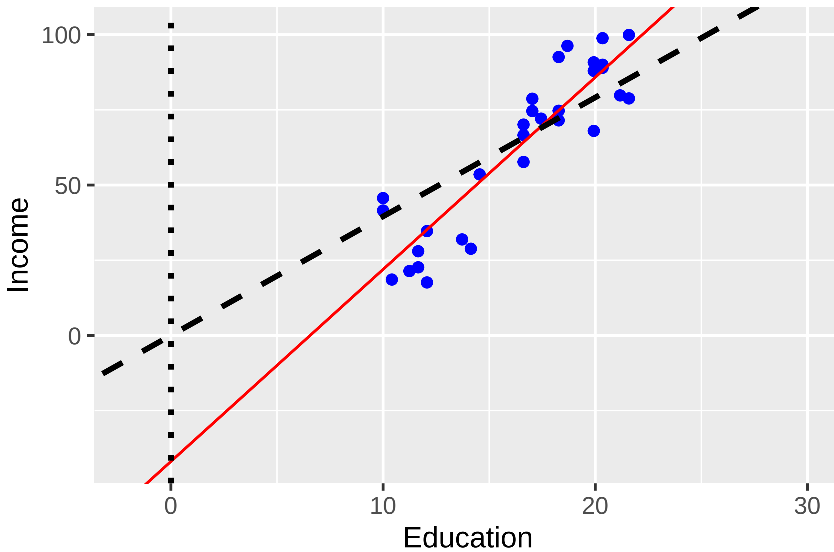
```
## Education
##  3.956202
```

```r
p_edu<-ggplot(inc, aes(y=Income, x=Education))+
  geom_point(color="blue")+
  geom_abline(intercept=coef(m_edu)[1],slope=coef(m_edu)[2],
              color="red")+
  geom_abline(intercept=0,slope=coef(m_edu_noint)[1],
              color="black", linetype="dashed", size=1)
p_edu
```



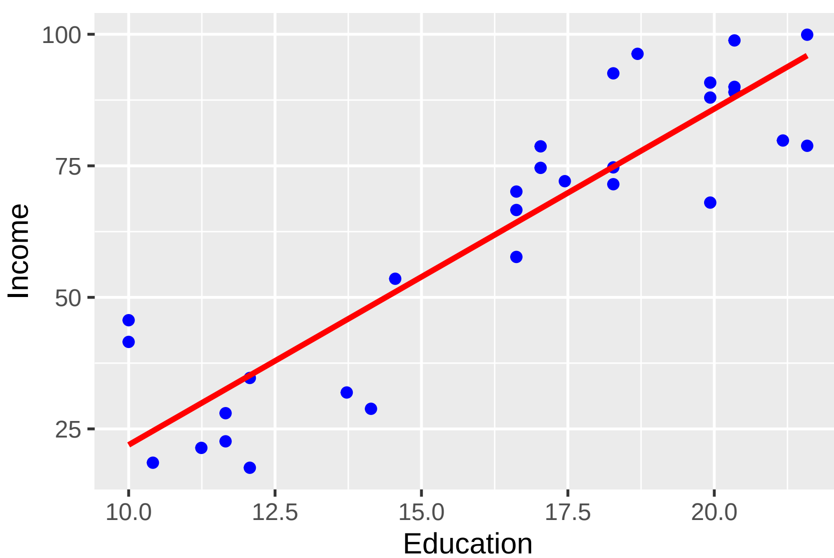

```r
# Expand axes so intercepts appear in plot and add vertical line at intercepts
p_edu<-ggplot(inc, aes(y=Income, x=Education))+
  geom_point(color="blue")+
  geom_abline(intercept=coef(m_edu)[1],slope=coef(m_edu)[2],
              color="red")+
  geom_abline(intercept=0,slope=coef(m_edu_noint)[1],
              color="black", linetype="dashed", size=1)+
  geom_vline(xintercept = 0, linetype="dotted", size=1)+
  xlim(-2,30)+
  ylim(-42,102)

p_edu
```



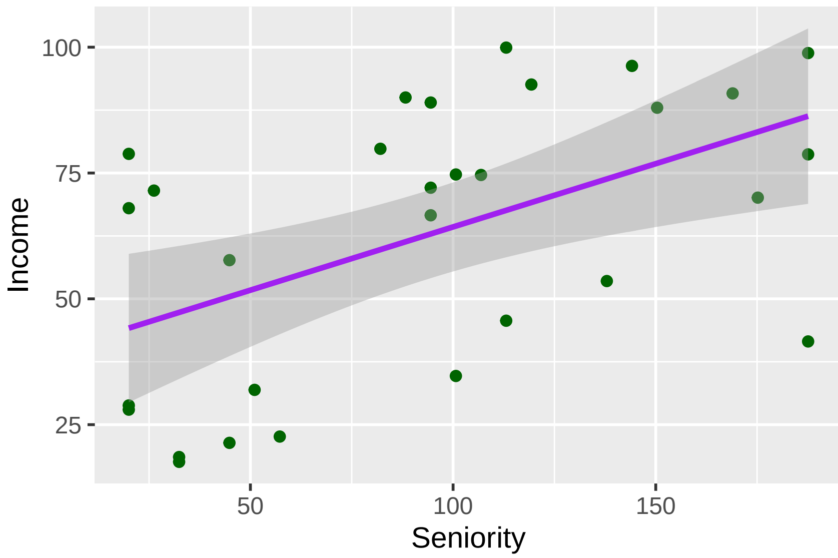
- `geom_smooth(method='lm')` adds a regression line with an intercept

```
ggplot(inc, aes(y=Income, x=Education))+
  geom_point(color="blue")+
  geom_smooth(method='lm', color="red", se=FALSE)
```



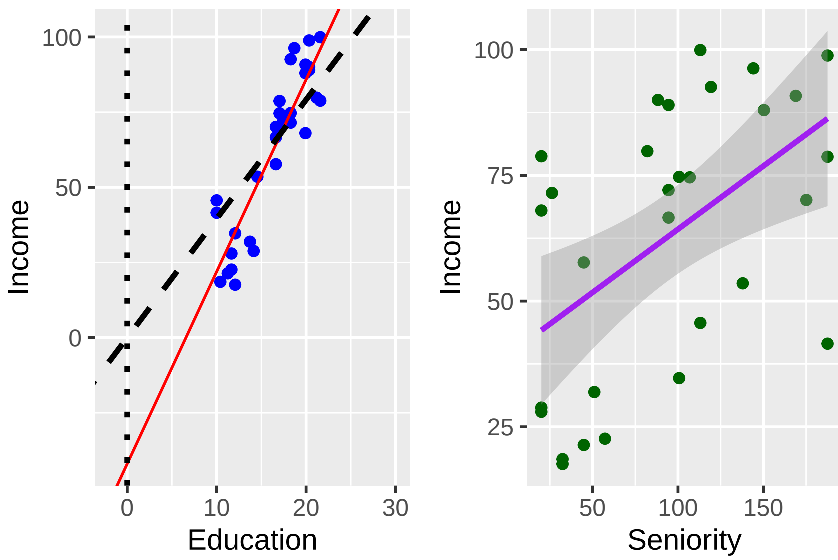
C. `Income` as a function of `Seniority`

```
p_sen<-ggplot(inc, aes(y=Income, x=Seniority))+
  geom_point(color="darkgreen")+
  geom_smooth(method='lm', color="purple", se=TRUE, size=1)
p_sen
```



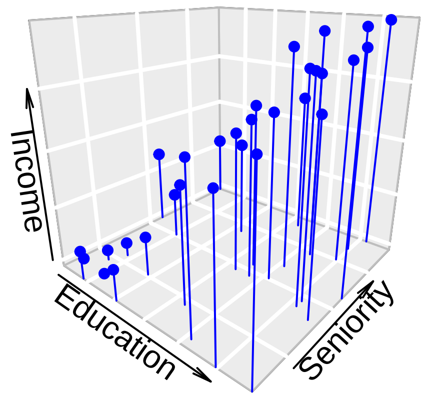
D. Putting `p_edu` and `p_sen` together

```
library(gridExtra)
library(grid)
grid.arrange(p_edu, p_sen, ncol = 2)
```



E. `Income` as a function of `Education` and `Seniority`

```
# install.packages("plot3D")
library(plot3D)
scatter3D(inc$Education, inc$Seniority, inc$Income, colvar=NULL,
          phi = 20, bty = "g", type="h", pch = 20, cex = 1, col="blue",
          xlab="Education", ylab="Seniority",zlab="Income")
```



- With the predicted surface

```
x <- inc$Education
y <- inc$Seniority
z <- inc$Income

# Compute the linear regression (z = b0+b1x+b2y+e)
pred_both <- lm(z ~ x + y)

# predict values on regular xy grid
grid.lines <- dim(inc)[[1]]
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid( x = x.pred, y = y.pred)
z.pred <- matrix(predict(pred_both, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)

fitpoints <- predict(pred_both)
scatter3D(x, y, z, colvar=NULL,
          phi=20, theta=20, bty ="g", type="h", pch=20, cex=1, col="blue",
          xlab="Education", ylab="Seniority", zlab="Income",
          surf = list(x=x.pred, y=y.pred, z=z.pred,
                 facets=NA, fit=fitpoints, col="red"))
```