Understanding `factors` in `R` Practice

Ryan Safner

11/19/2018

Let's generate some random data. We will make a `data.frame` called `df` (feel free to call it something else.)

```
set.seed(1) # this makes data reproducible, if we set the same seed #, we all get the same "random" dat

colors<-c("red", "orange","yellow", "green", "blue", "purple")
color<-sample(colors,100,replace=TRUE)

educ.levels<-c("high school","college", "graduate degree")
education<-sample(educ.levels, 100, replace=TRUE)


# make a data frame of three variables, x, y, and color
df<-data.frame(x=rnorm(100,2,1), # x is 100 draws from a normal distr with mean 2 and sd 1
               y=rnorm(100,5,1), # y is 100 draws from a normal distr with mean 5 and sd 1
               color=factor(color), # color is a factor
               education=ordered(education, levels=c("high school", "college", "graduate degree")) #
             )
```

1. Get a summary of `df`. Also check its `str()` and `head()` to get a closer look. Look at the data itself with `View(df)`.

```
summary(df) # get summary of data
```

```
##        x                y            color              education
##  Min.   :0.08564   Min.   :2.111   blue  :21   high school    :29
##  1st Qu.:1.34895   1st Qu.:4.545   green :13   college        :40
##  Median :1.82278   Median :4.998   orange:16   graduate degree:31
##  Mean   :1.96219   Mean   :5.030   purple:14
##  3rd Qu.:2.50090   3rd Qu.:5.698   red   :11
##  Max.   :4.30798   Max.   :7.649   yellow:25
```

```
str(df)
```

```
## 'data.frame':    100 obs. of  4 variables:
##  $ x        : num  1.38 2.04 1.09 2.16 1.35 ...
##  $ y        : num  5.41 6.69 6.59 4.67 2.71 ...
##  $ color    : Factor w/ 6 levels "blue","green",..: 3 6 2 4 3 4 4 2 2 5 ...
##  $ education: Ord.factor w/ 3 levels "high school"<..: 2 2 1 3 2 1 1 2 3 2 ...
```

```
head(df)
```

```
##          x        y  color       education
## 1 1.379633 5.409402 orange         college
## 2 2.042116 6.688873 yellow         college
## 3 1.089078 6.586588  green     high school
## 4 2.158029 4.669092 purple graduate degree
## 5 1.345415 2.714764 orange         college
## 6 3.767287 7.497662 purple     high school
```

```
#View(df)
```

2. Look more closely at `color`. Check its `class()`, `nlevels()` and the actual `levels()`. Finally, make a `table()` of the counts of each category.

```
class(df$color)
```

```
## [1] "factor"
```

```
nlevels(df$color)
```

```
## [1] 6
```

```
levels(df$color)
```

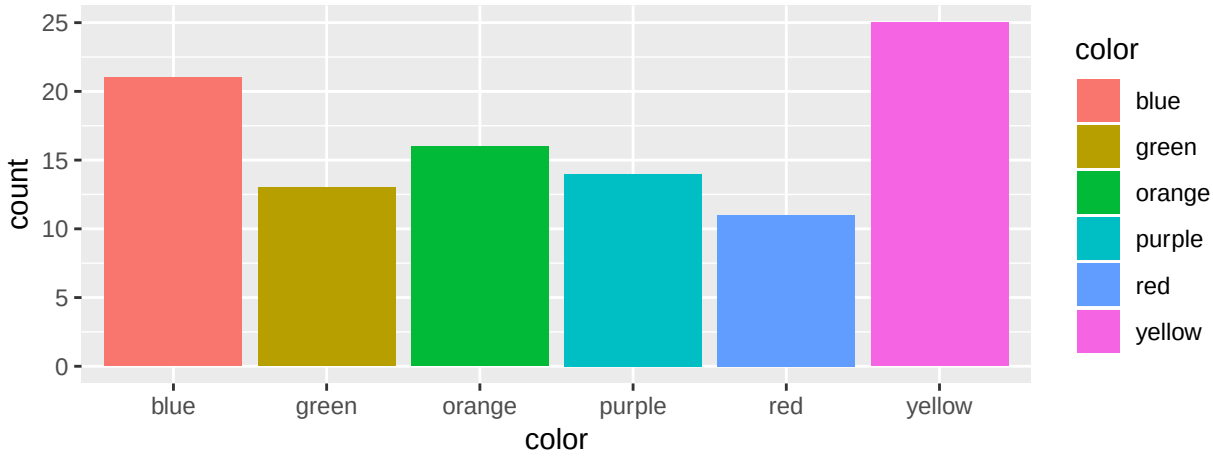
```
## [1] "blue"   "green"  "orange" "purple" "red"    "yellow"
```

```
table(df$color) # get counts of color
```

```
##
##   blue  green orange purple    red yellow
##     21     13     16     14     11     25
```

3. Make a barplot of `color`.

```
library(ggplot2)

ggplot(data = df, aes(x = color))+
  geom_bar(aes(fill=color))
```



4. Look more closely at `education`. Check its `class()`, `nlevels()` and the actual `levels()`. Finally, make a `table()` of the counts of each category.

```
class(df$education)
```

```
## [1] "ordered" "factor"
```

```r
nlevels(df$education)
```

```
## [1] 3
```

```r
levels(df$education)
```

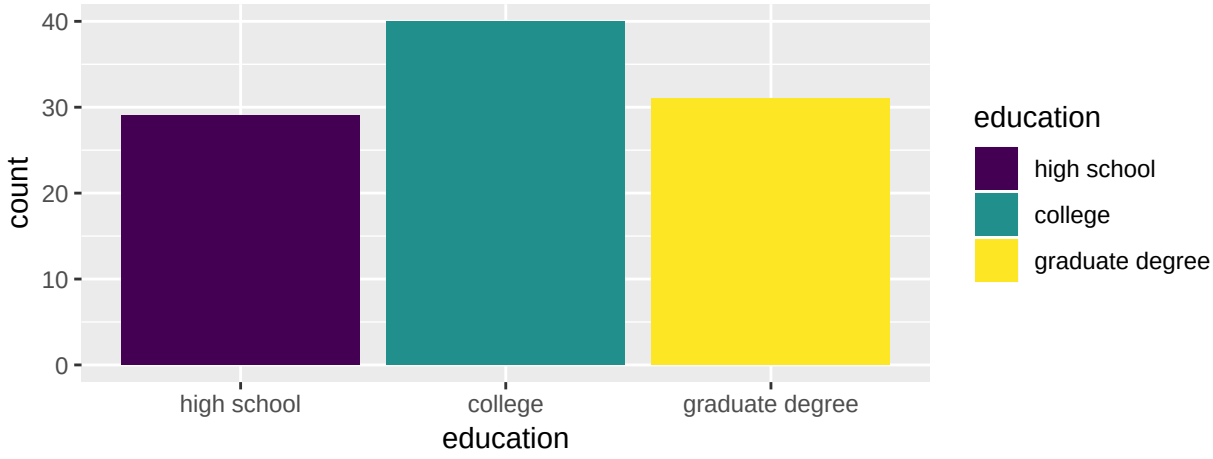
```
## [1] "high school"     "college"         "graduate degree"
```

```r
table(df$education) # get counts of color
```

```
##
##     high school         college graduate degree
##              29              40              31
```

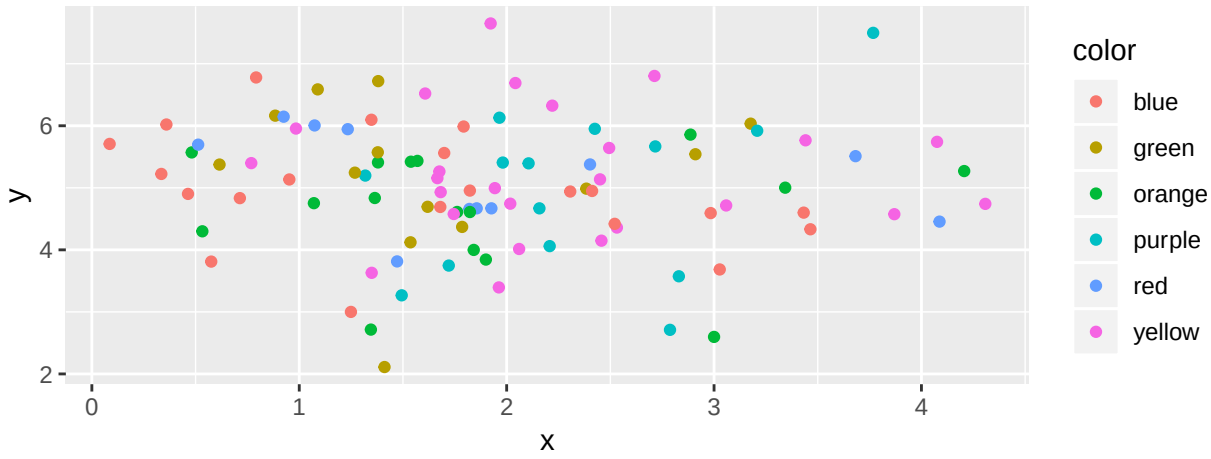
5. Make a barplot of `education`.

```r
ggplot(data = df, aes(x = education))+
  geom_bar(aes(fill=education))
```



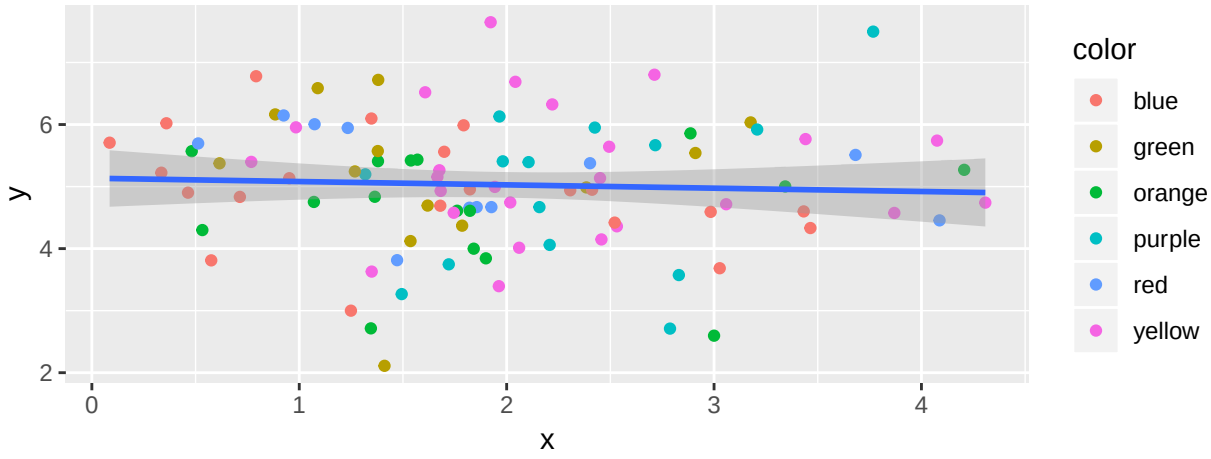
6. Now let's try looking at plots by different categories. Make a scatterplot of `x` and `y` and add `color=color` to your base layer `aes()`.

```r
ggplot(data = df, aes(x = x, y = y, color=color))+
  geom_point()
```



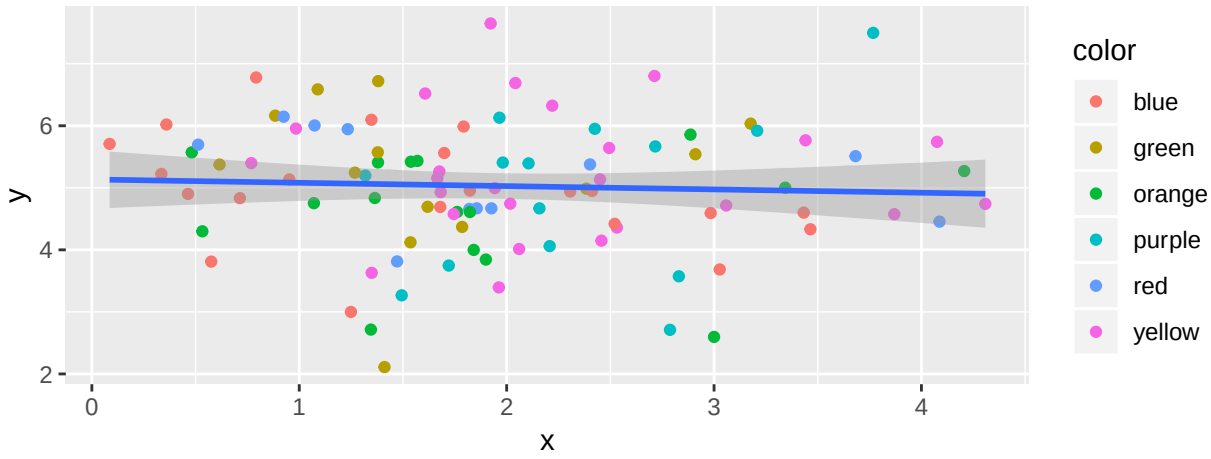
a. Now let's try subsetting. Plot only data for the color `green`.

```
ggplot(data = df, aes(x = x, y = y))+
  geom_point(aes(color=color))+
  geom_smooth(method="lm")
```



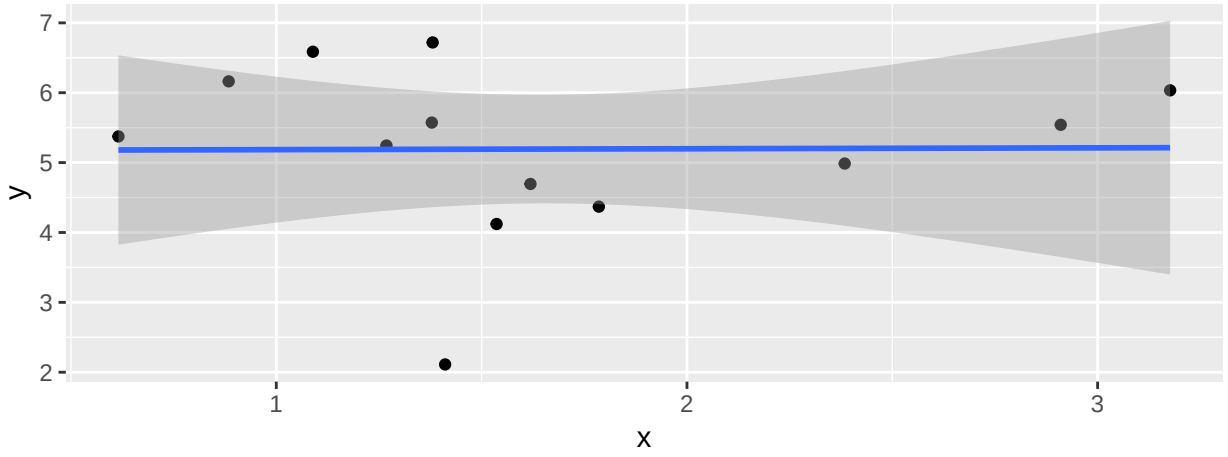
b. In addition to your `geom_point()`, add a `geom_smooth(method="lm")` regression line. Notice it makes a regression line for each color. If we want an overall regression line, we need to redo our scatterplot as follows. In the base layer, don't include `color` in your `aes()`, move it instead inside `geom_point(aes(color=color))`. Then add a `geom_smooth()`, it will do it for the overall plot.

```
ggplot(data = df, aes(x = x, y = y))+
  geom_point(aes(color=color))+
  geom_smooth(method="lm")
```

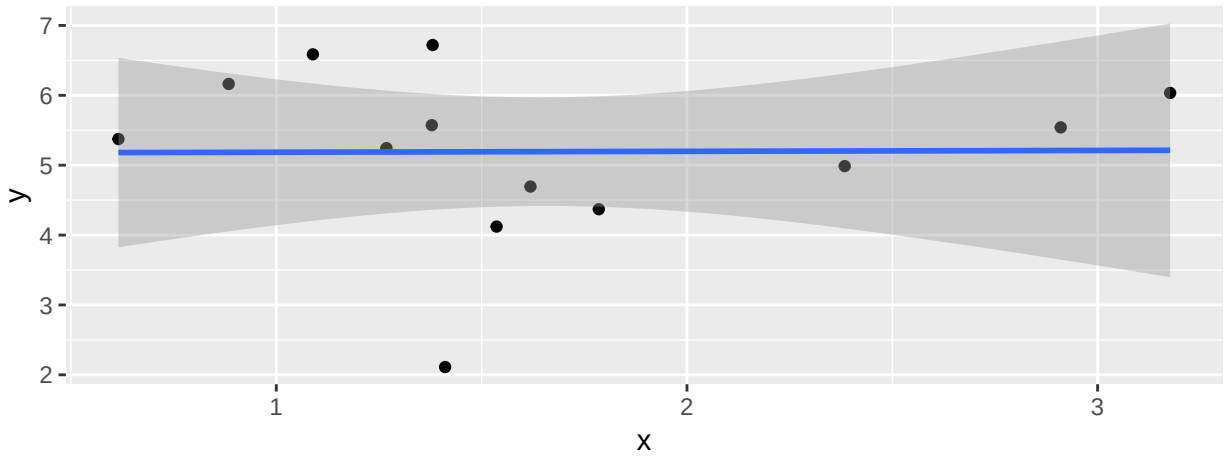


c. Now let's try subsetting. Make a scatterplot and regression line only with data points for the color green.

```
# two ways to do this, one using the index [row, columns] selecting only rows for which color is green

ggplot(data = df[df$color=="green",], aes(x = x, y = y))+
  geom_point()+
  geom_smooth(method="lm")
```

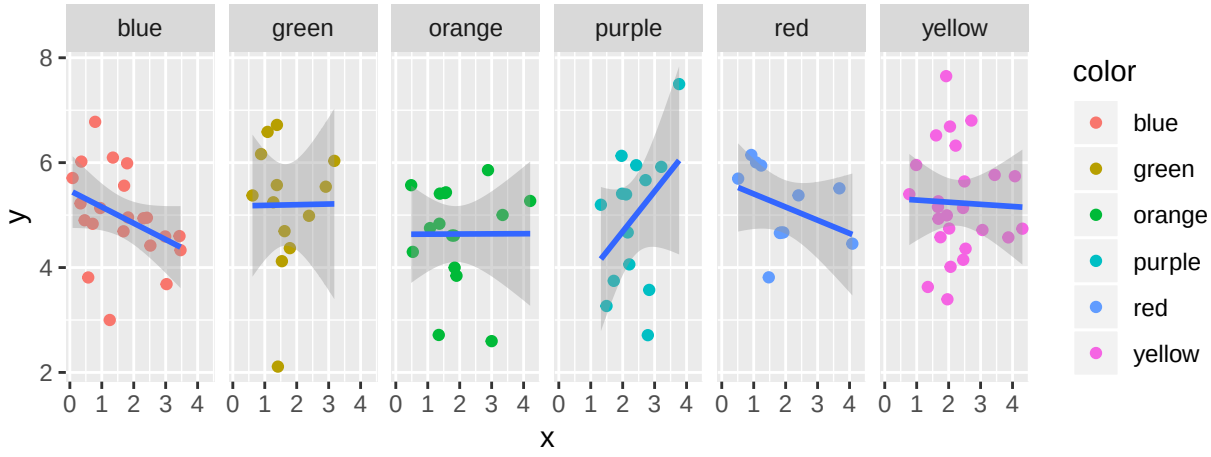



```
# another is to use subset and make another df to use

df.green<-subset(df,color=="green")

ggplot(data = df.green, aes(x = x, y = y))+
  geom_point()+
  geom_smooth(method="lm")
```



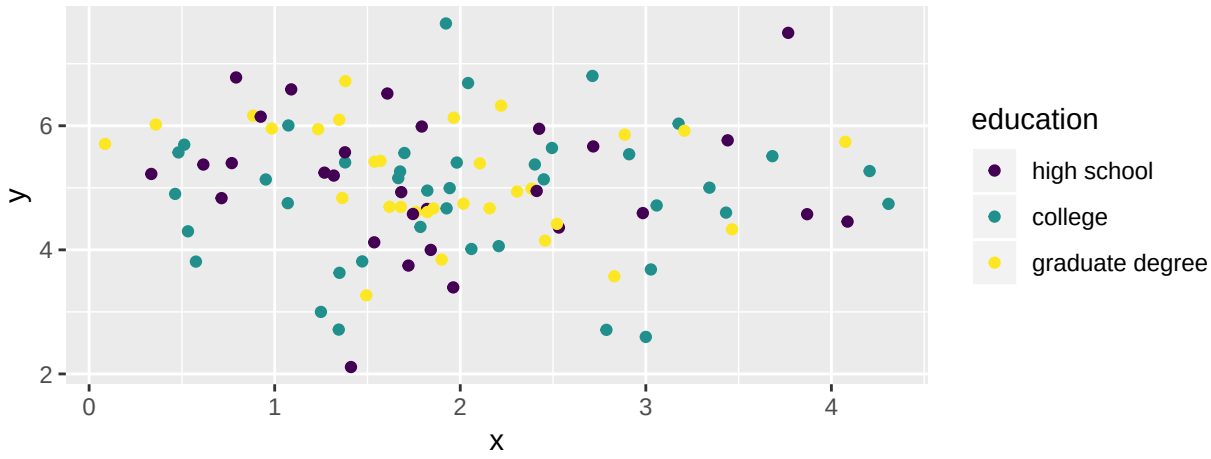
d. Let's simply use the `facet_grid()` command to plot all the different colors as different plots. Reuse your commands from the first plot in this question, and then add a `facet` layer with `+facet_grid(cols=vars(color))`

```
ggplot(data = df, aes(x = x, y = y))+
  geom_point(aes(color=color))+
  geom_smooth(method="lm")+
  facet_grid(cols=vars(color))
```

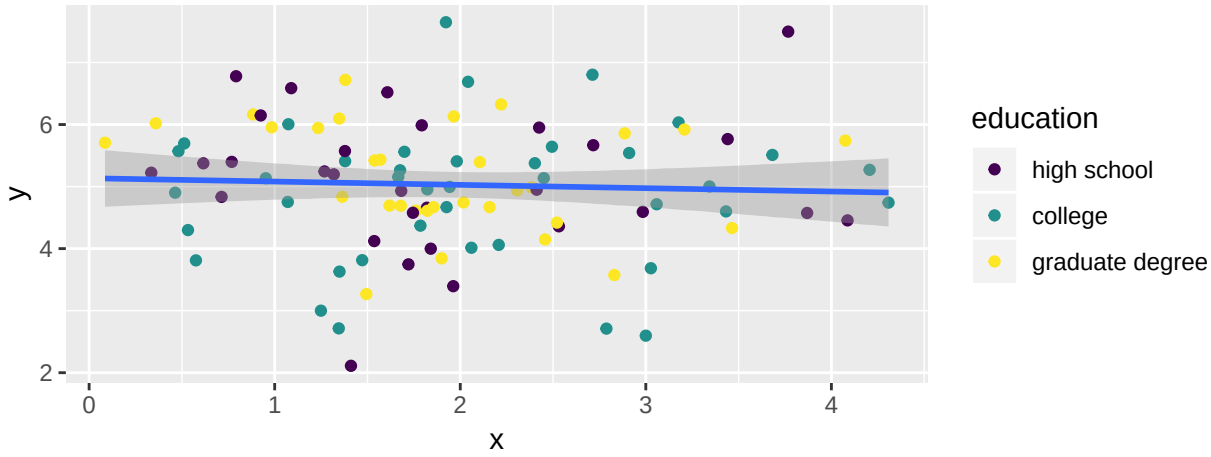


7. Run through problem #6 again, but using `education` instead of `color`.

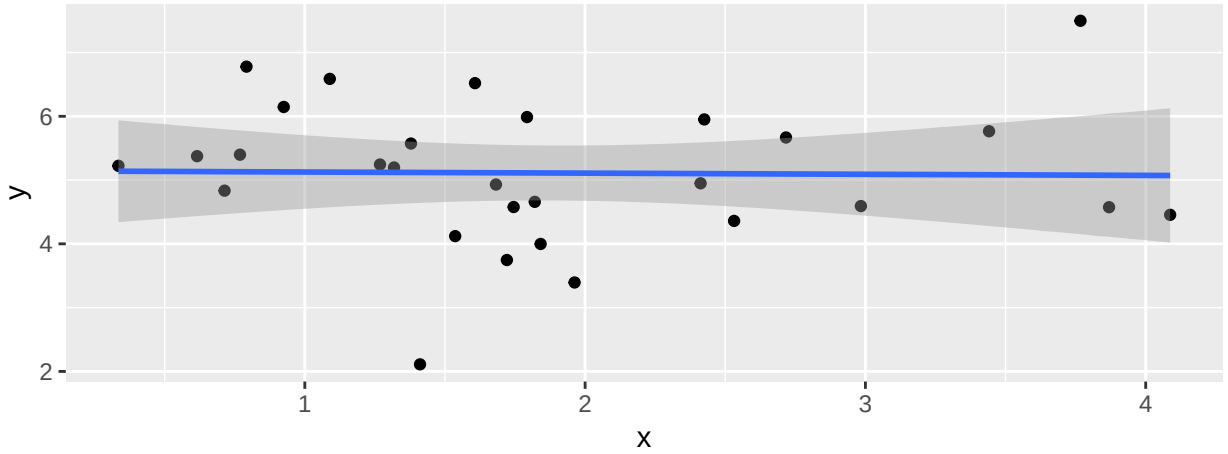
```
ggplot(data = df, aes(x = x, y = y, color=education))+
  geom_point()
```



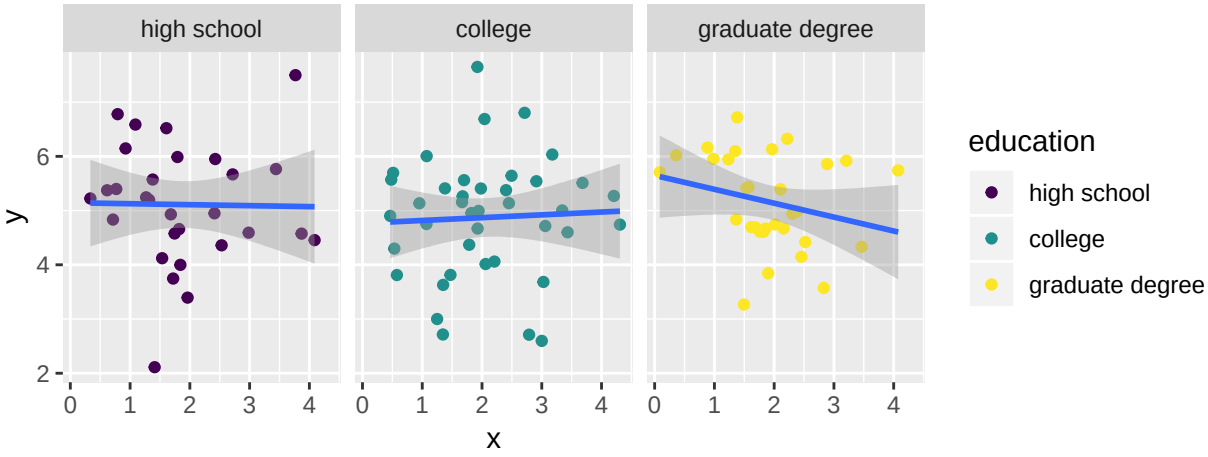

```
ggplot(data = df, aes(x = x, y = y))+
  geom_point(aes(color=education))+
  geom_smooth(method="lm")
```




```
df.hs<-subset(df,education=="high school")

ggplot(data = df.hs, aes(x = x, y = y))+
  geom_point()+
  geom_smooth(method="lm")
```



```
ggplot(data = df, aes(x = x, y = y))+
  geom_point(aes(color=education))+
  geom_smooth(method="lm")+
  facet_grid(cols=vars(education))
```



8. Now let's try some regression.

a. Run a regression of y on x and education. What happens?

```
reg<-lm(y~x+education, data=df)
summary(reg)
```

```
##
## Call:
## lm(formula = y ~ x + education, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02020 -0.55160  0.06745  0.71197  2.77124
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.13132    0.23830  21.533   <2e-16 ***
## x           -0.04338    0.10959  -0.396    0.693
## education.L  0.03367    0.19020   0.177    0.860
## education.Q  0.20817    0.17410   1.196    0.235
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 96 degrees of freedom
## Multiple R-squared:  0.01751,    Adjusted R-squared:  -0.01319
## F-statistic: 0.5703 on 3 and 96 DF,  p-value: 0.6359
```

b. `R` was generous and did the work for you! But let's do the same thing ourselves manually. What we need to do is convert `education` into a three dummy variables, one for each level of education. It's easiest to use the `ifelse()` command here. Remember the syntax: `ifelse(condition, do.this.if.true, do.this.if.false)`. Check your data `df` again with `head()` or `View()` to make sure you properly coded the variables.

```
df$hs<-ifelse(education=="high school",1,0)
df$college<-ifelse(education=="college",1,0)
df$grad<-ifelse(education=="graduate degree",1,0)
```

```
head(df)
```

```
##          x        y  color       education hs college grad
## 1 1.379633 5.409402 orange         college  0       1    0
## 2 2.042116 6.688873 yellow         college  0       1    0
## 3 1.089078 6.586588  green     high school  1       0    0
## 4 2.158029 4.669092 purple graduate degree  0       0    1
## 5 1.345415 2.714764 orange         college  0       1    0
## 6 3.767287 7.497662 purple     high school  1       0    0
```

c. Now run a regression of `y` on `x` and all of your new dummy variables. What happens, and why?

```
reg2<-lm(y~x+hs+college+grad, data=df)
summary(reg2)
```

```
##
## Call:
## lm(formula = y ~ x + hs + college + grad, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.02020 -0.55160  0.06745  0.71197  2.77124
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.24011    0.28138  18.623   <2e-16 ***
## x           -0.04338    0.10959  -0.396    0.693
## hs          -0.04762    0.26898  -0.177    0.860
## college     -0.27877    0.24956  -1.117    0.267
## grad             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 96 degrees of freedom
## Multiple R-squared:  0.01751,    Adjusted R-squared:  -0.01319
## F-statistic: 0.5703 on 3 and 96 DF,  p-value: 0.6359
```

d. Run three different regressions, each one omitting one of the different categories of education. Interpret your coefficients.

```
reg.no.hs<-lm(y~x+college+grad, data=df)
reg.no.college<-lm(y~x+hs+grad, data=df)
reg.no.grad<-lm(y~x+hs+college, data=df)

suppressPackageStartupMessages(library("stargazer"))
stargazer(reg.no.hs, reg.no.college, reg.no.grad, type="latex", float=F, header=F)
```

	Dependent variable:		
	y		
	(1)	(2)	(3)
x	−0.043	−0.043	−0.043
	(0.110)	(0.110)	(0.110)
college	−0.231		−0.279
	(0.255)		(0.250)
hs		0.231	−0.048
		(0.255)	(0.269)
grad	0.048	0.279	
	(0.269)	(0.250)	
Constant	5.192***	4.961***	5.240***
	(0.283)	(0.279)	(0.281)
Observations	100	100	100
R^2	0.018	0.018	0.018
Adjusted R^2	−0.013	−0.013	−0.013
Residual Std. Error (df = 96)	1.041	1.041	1.041
F Statistic (df = 3; 96)	0.570	0.570	0.570
Note:	*p<0.1; **p<0.05; ***p<0.01		