Lecture 13

Quantitative Political Science

Prof. Bisbee

Vanderbilt University

Lecture Date: 2023/10/24

Slides Updated: 2023-10-20

Agenda

1. Associations
2. Describing relationships
3. Modeling relationships

Associations between two (or more) variables

- Thus far, description & inference for either one variable or two variables drawn from different units
- Now, what you are here for! Relationships between two variables from the *same* units!
- Bivariate:
  - X (independent / predictor / explanatory / RHS) and Y (dependent / outcome)
  - Implies a causal intuition, but does NOT buy it!
- 4 approaches (more like steps)
  1. Displaying relationships
  2. Summarizing **non-parametrically**
  3. Summarizing **parametrically**
  4. Making **inferences** about the relationship in a population from a sample

1. Displaying Relationships

- Crosstab(ulation)s: Y typically in rows, X in columns

```
require(tidyverse)
dat <- read_rds('https://github.com/jbisbee1/PSCI_8356/raw/main/Lectures/Data/sc_debt.Rds')

t <- table(dat$preddeg,dat$control)
t
```

```
##
##               Private Public
##   Associate       127    694
##   Bachelor's     1193    532
```

1. Displaying Relationships

- Crosstab(ulation)s: Y typically in rows, X in columns

```
prop.table(t,margin = 1) # Rows
```

```
##
##                 Private    Public
##   Associate  0.1546894 0.8453106
##   Bachelor's 0.6915942 0.3084058
```

```
prop.table(t,margin = 2) # Columns
```

```
##
##                  Private     Public
##   Associate  0.09621212 0.56606852
##   Bachelor's 0.90378788 0.43393148
```

1. Displaying Relationships

- Crosstab(ulation)s: Y typically in rows, X in columns...BAD FOR MANY CATEGORIES OR CONTINUOUS!

```
table(dat$md_earn_wne_p6,dat$sat_avg)
```

```
## 
##          737 847 849 851 854 855 861 865 875 876 877 880
##   10600    0   0   0   0   0   0   0   0   0   0   0   0
##   11000    0   0   0   0   0   0   0   0   0   0   0   0
##   11800    0   0   0   0   0   0   0   0   0   0   0   0
##   11900    0   0   0   0   0   0   0   0   0   0   0   0
##   12200    0   0   0   0   0   0   0   0   0   0   0   0
##   12800    0   0   0   0   0   0   0   0   0   0   0   0
##   12900    0   0   0   0   0   0   0   0   0   0   0   0
##   13000    0   0   0   0   0   0   0   0   0   0   0   0
##   13400    0   0   0   0   0   0   0   0   0   0   0   0
##   13700    0   0   0   0   0   0   0   0   0   0   0   0
##   14000    0   0   0   0   0   0   0   0   0   0   0   0
##   14100    0   0   0   0   0   0   0   0   0   0   0   0
##   14200    0   0   0   0   0   0   0   0   0   0   0   0
##   14300    0   0   0   0   0   0   0   0   0   0   0   0
##   14400    0   0   0   0   0   0   0   0   0   0   0   0
##   14500    0   0   0   0   0   0   0   0   0   0   0   0
```

1. Displaying Relationships

- Use bins first

```
dat <- dat %>%
  mutate(earn_quartiles = cut(md_earn_wne_p6,breaks = quantile(md_earn_wne_p6,p =
c(0,.25,.5,.75,1),na.rm=T),dig.lab = 10),
         sat_quartiles = cut(sat_avg,breaks = quantile(sat_avg,p = c(0,.25,.5,.75,1),na.rm=T),dig.lab =
10))
t <- table(dat$earn_quartiles,dat$sat_quartiles)
round(prop.table(t,margin = 2)*100,2)
```

```
##
##                  (737,1053] (1053,1119] (1119,1205]
##   (10600,26100]       20.33        3.68        0.66
##   (26100,31500]       34.00       27.76       19.87
##   (31500,37400]       31.33       50.17       39.40
##   (37400,120400]      14.33       18.39       40.07
##
##                  (1205,1557]
##   (10600,26100]         2.03
##   (26100,31500]         5.07
##   (31500,37400]        22.97
##   (37400,120400]       69.93
```

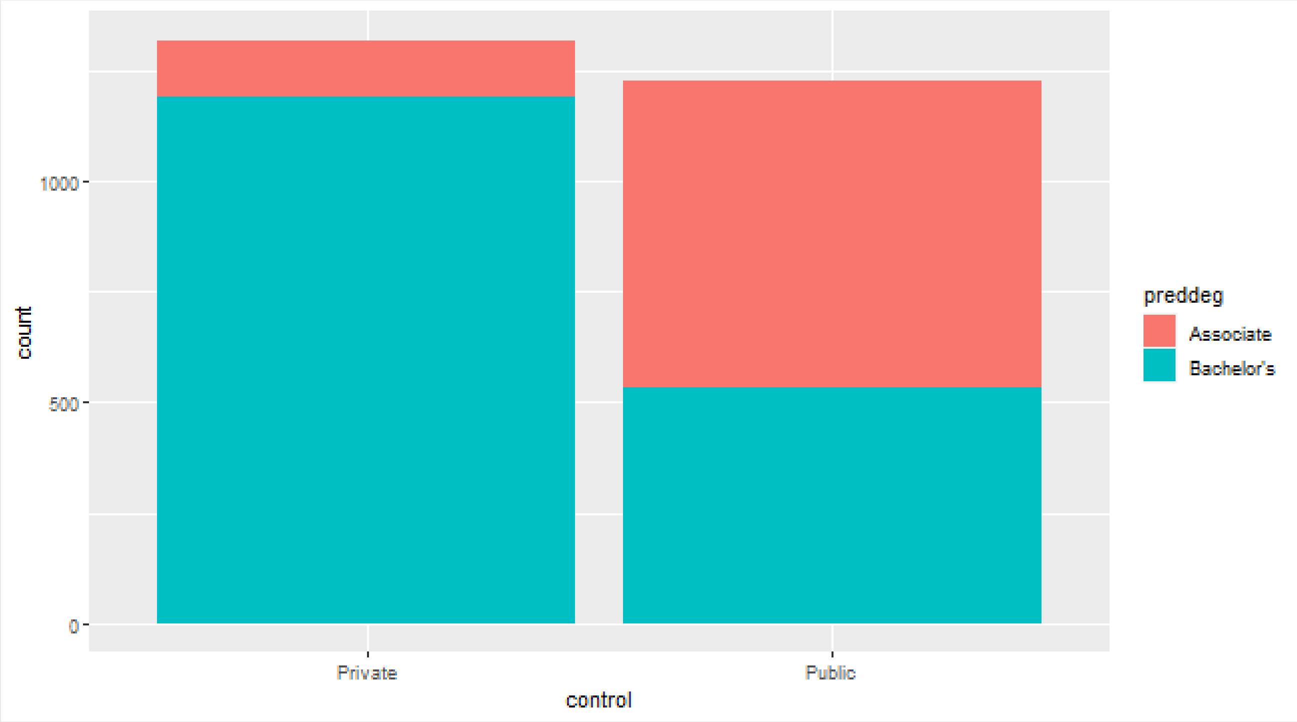
1. Displaying Relationships

- Plotting
  - Barplots (`geom_bar()`): X and Y are both categorical (including binary)
  - Densities / histograms (`geom_density()` / `geom_histogram()`): X is binary and Y is continuous
  - Boxplots / violin plots (`geom_boxplot()` / `geom_violin()`): X is categorical and Y is continuous
  - Scatterplots (`geom_point()`): X and Y are both continuous

1. Displaying Relationships

- Barplots (`geom_bar()`): X and Y are both categorical (including binary)

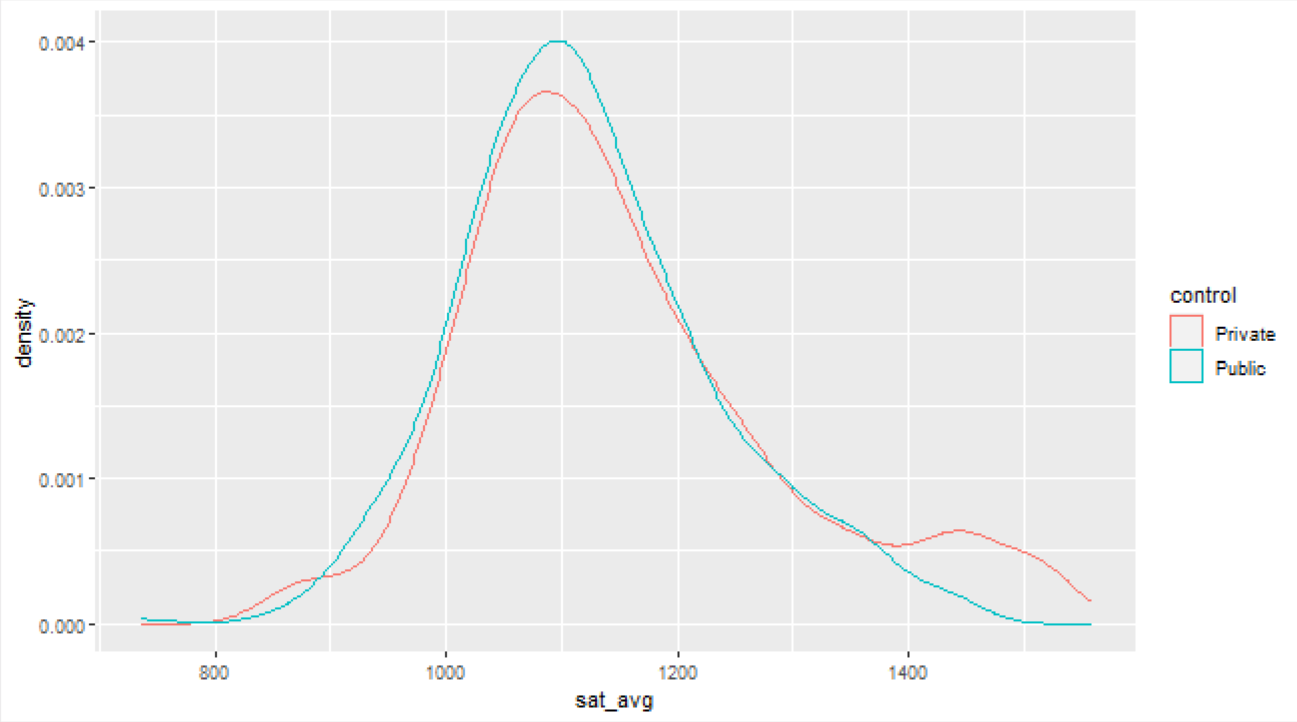
```
dat %>%
  ggplot(aes(x = control,fill = preddeg)) +
  geom_bar()
```



1. Displaying Relationships

- Densities / histograms (`geom_density()` / `geom_histogram()`): X is binary and Y is continuous

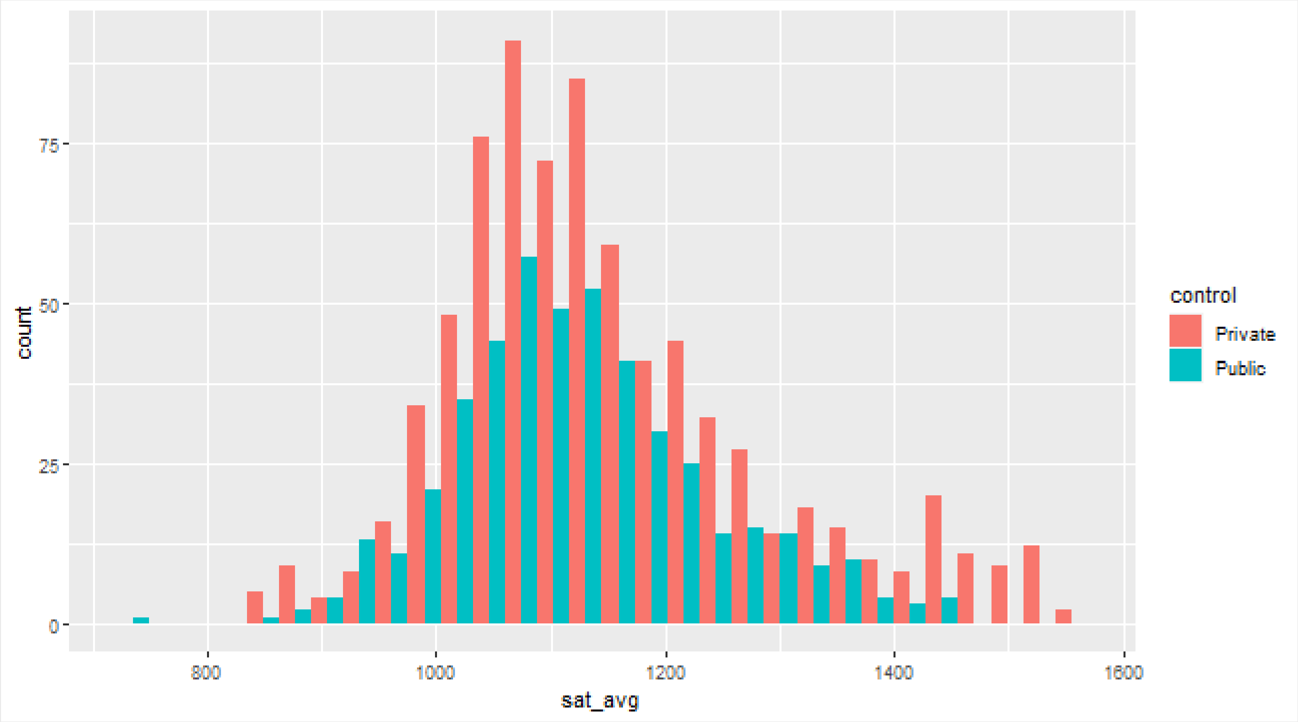
```
dat %>%
  ggplot(aes(x = sat_avg,color = control)) +
  geom_density()
```



1. Displaying Relationships

- Densities / histograms (`geom_density()` / `geom_histogram()`): X is binary and Y is continuous

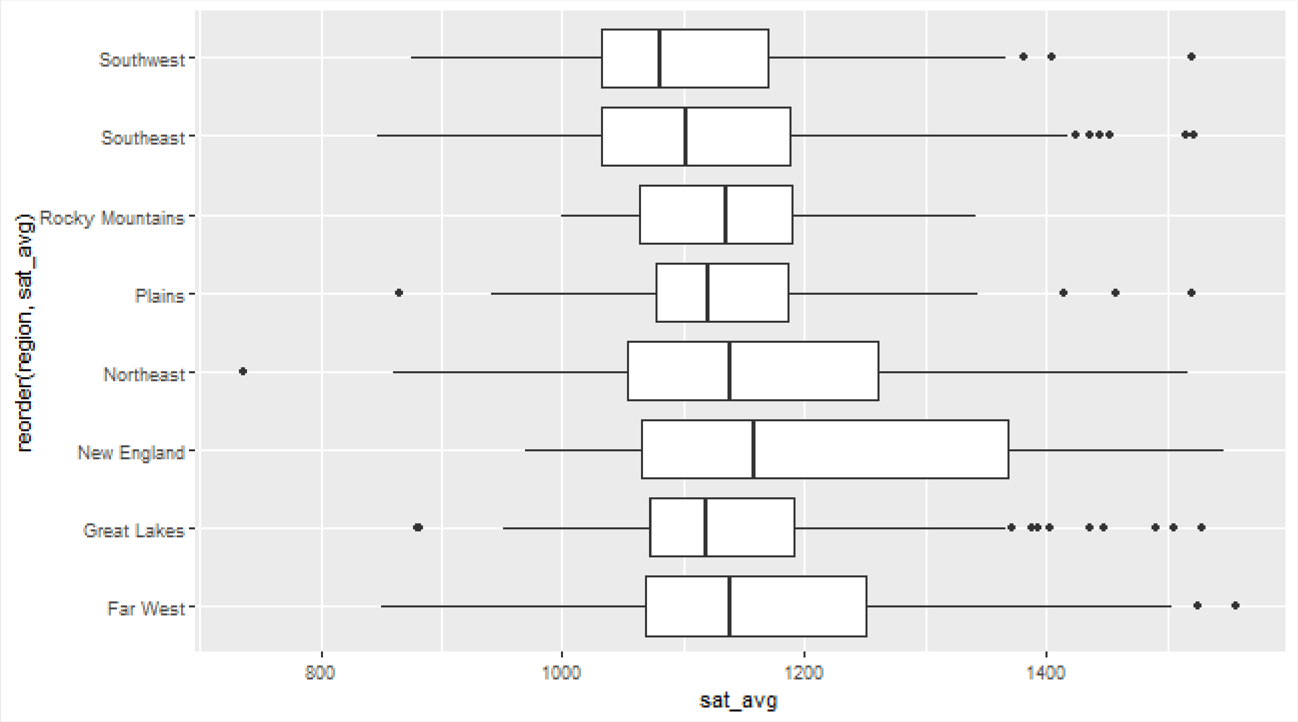
```
dat %>%
  ggplot(aes(x = sat_avg,fill = control)) +
  geom_histogram(position = 'dodge')
```



1. Displaying Relationships

- Boxplots / violin plots (`geom_boxplot()` / `geom_violin()`): X is categorical and Y is continuous

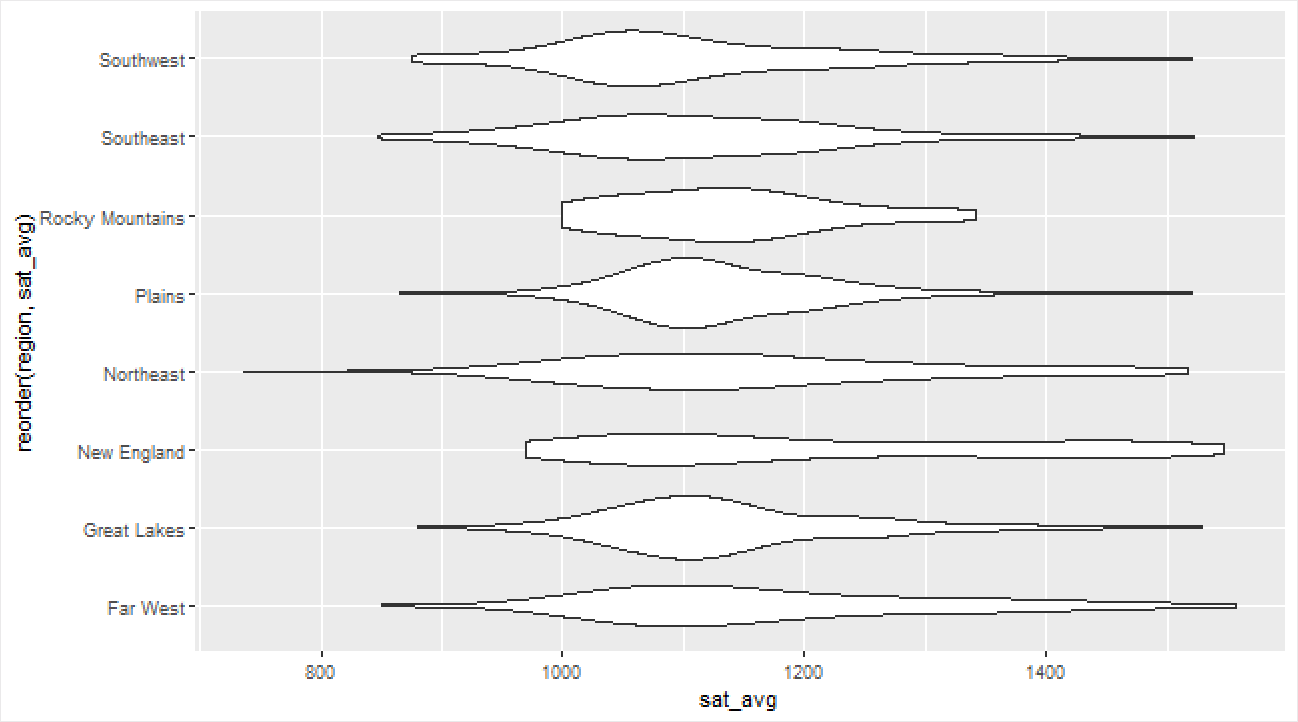
```
dat %>%
  ggplot(aes(x = sat_avg,y = reorder(region,sat_avg))) +
  geom_boxplot()
```



1. Displaying Relationships

- Boxplots / violin plots (`geom_boxplot()` / `geom_violin()`): X is categorical and Y is continuous

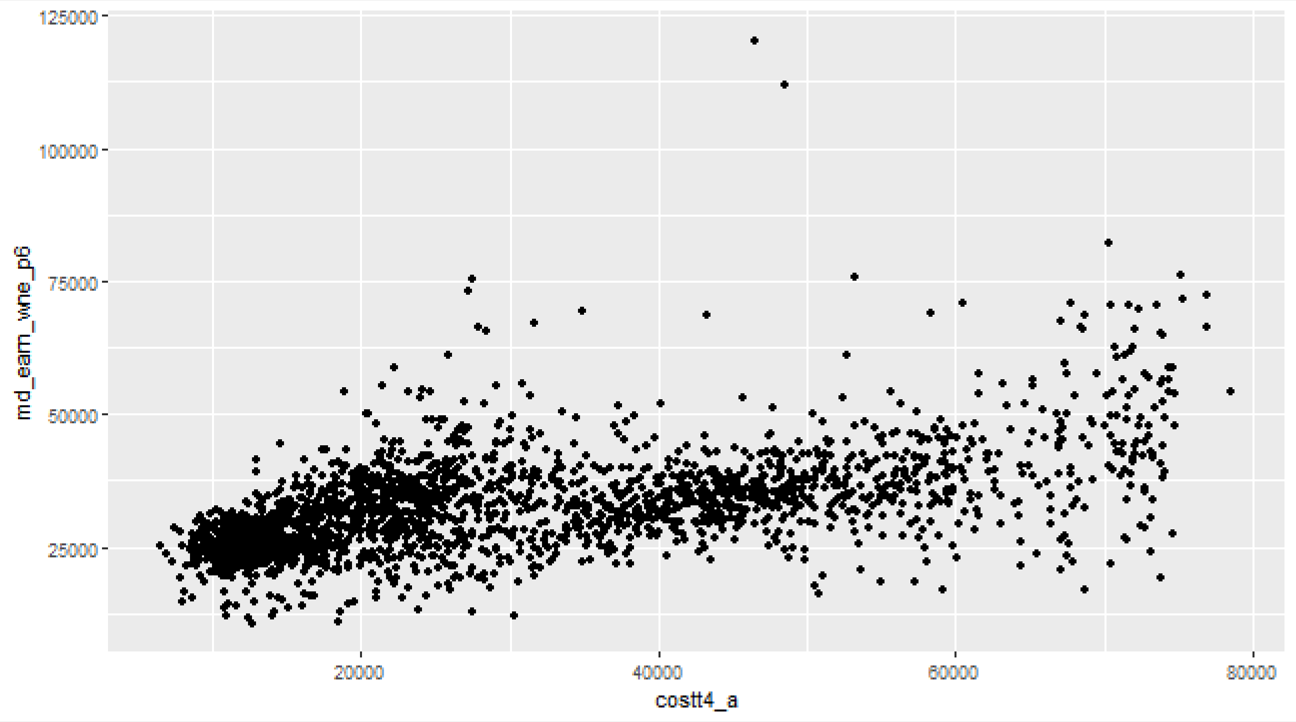
```
dat %>%
  ggplot(aes(x = sat_avg,y = reorder(region,sat_avg))) +
  geom_violin()
```



1. Displaying Relationships

- Scatterplots (`geom_point()`): X and Y are both continuous

```
dat %>%
  ggplot(aes(x = costt4_a,y = md_earn_wne_p6)) +
  geom_point()
```



2. Summarizing Non-parametrically

- Conditional means
  - What is the average value of Y for a given value of X?

```
dat %>%
  drop_na(sat_quartiles) %>%
  group_by(sat_quartiles) %>%
  summarise(mean_earn = mean(md_earn_wne_p6,na.rm=T))
```

```
## # A tibble: 4 × 2
##   sat_quartiles mean_earn
##   <fct>             <dbl>
## 1 (737,1053]       31118
## 2 (1053,1119]      34352.
## 3 (1119,1205]      36507.
## 4 (1205,1557]      44234.
```

2. Summarizing Non-parametrically

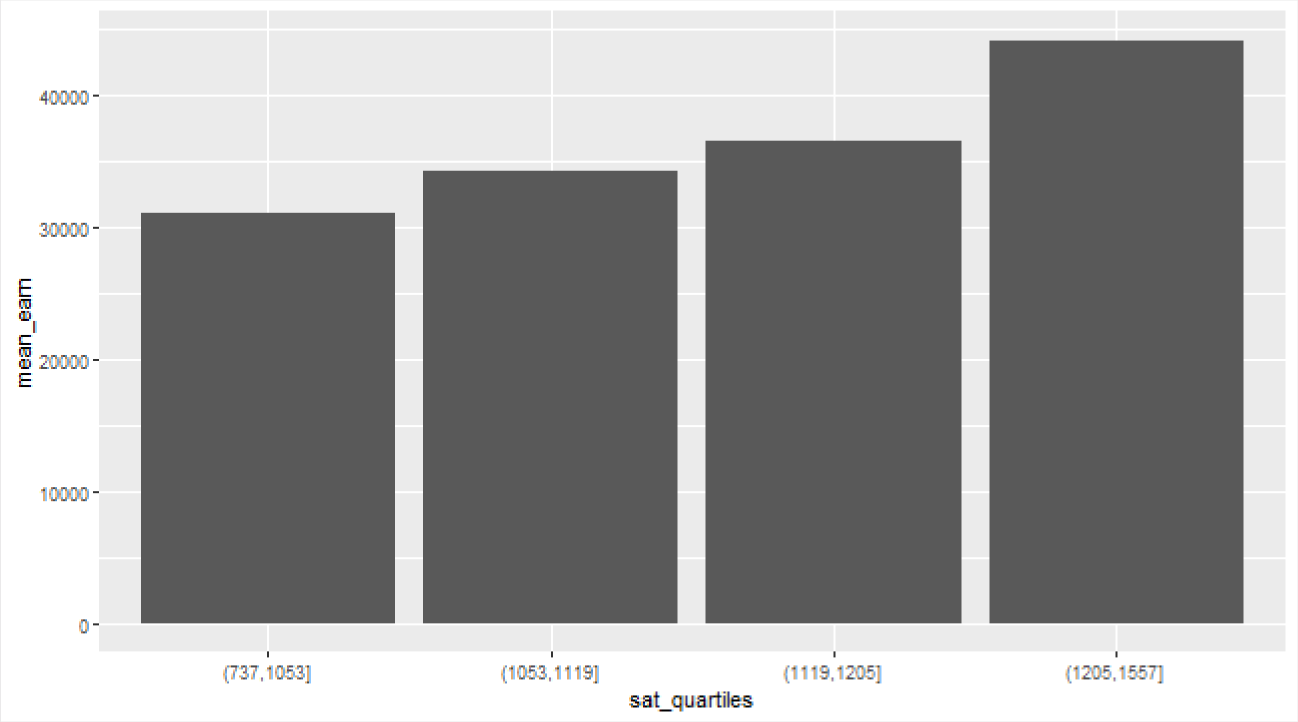
- Conditional means
    - What is the average value of Y for a given value of X?

```
p <- dat %>%
  drop_na(sat_quartiles) %>%
  group_by(sat_quartiles) %>%
  summarise(mean_earn = mean(md_earn_wne_p6,na.rm=T)) %>%
  ggplot(aes(x = sat_quartiles,y = mean_earn)) +
  geom_bar(stat = 'identity')
```

2. Summarizing Non-parametrically

- Conditional means
  - What is the average value of Y for a given value of X?

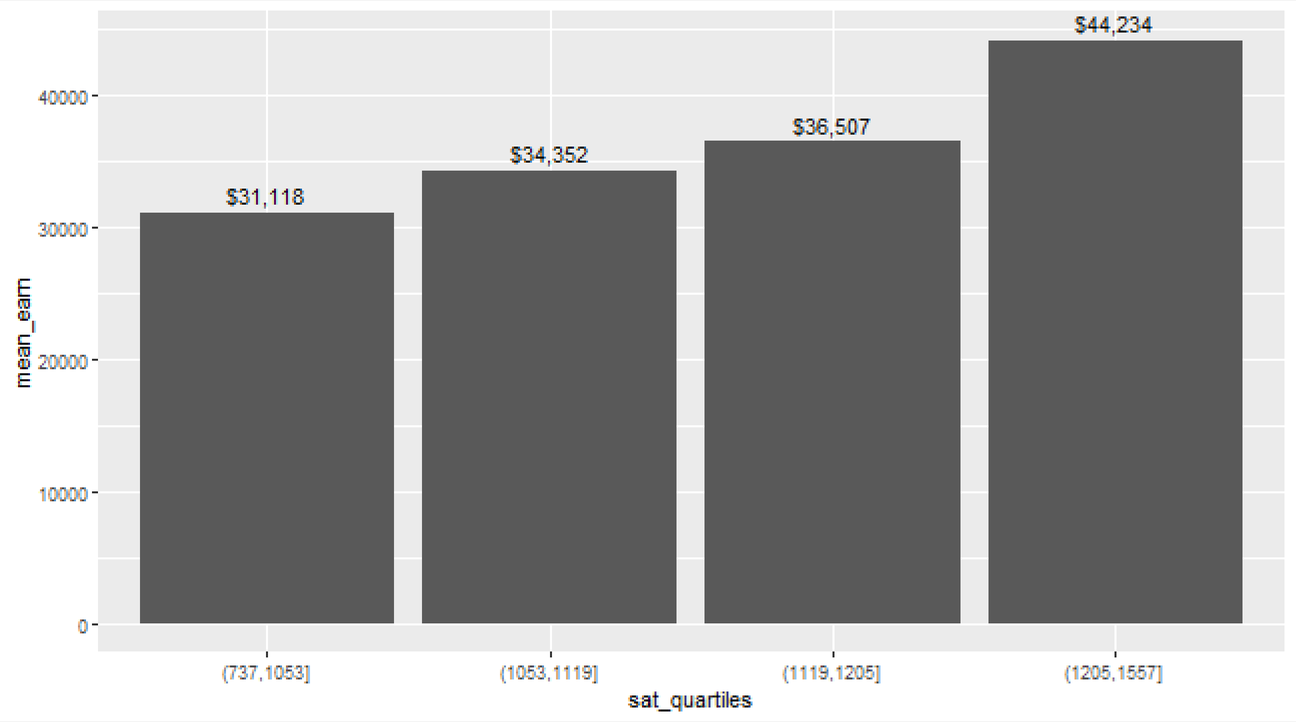
```
p
```



2. Summarizing Non-parametrically

- Conditional means
  - What is the average value of Y for a given value of X?

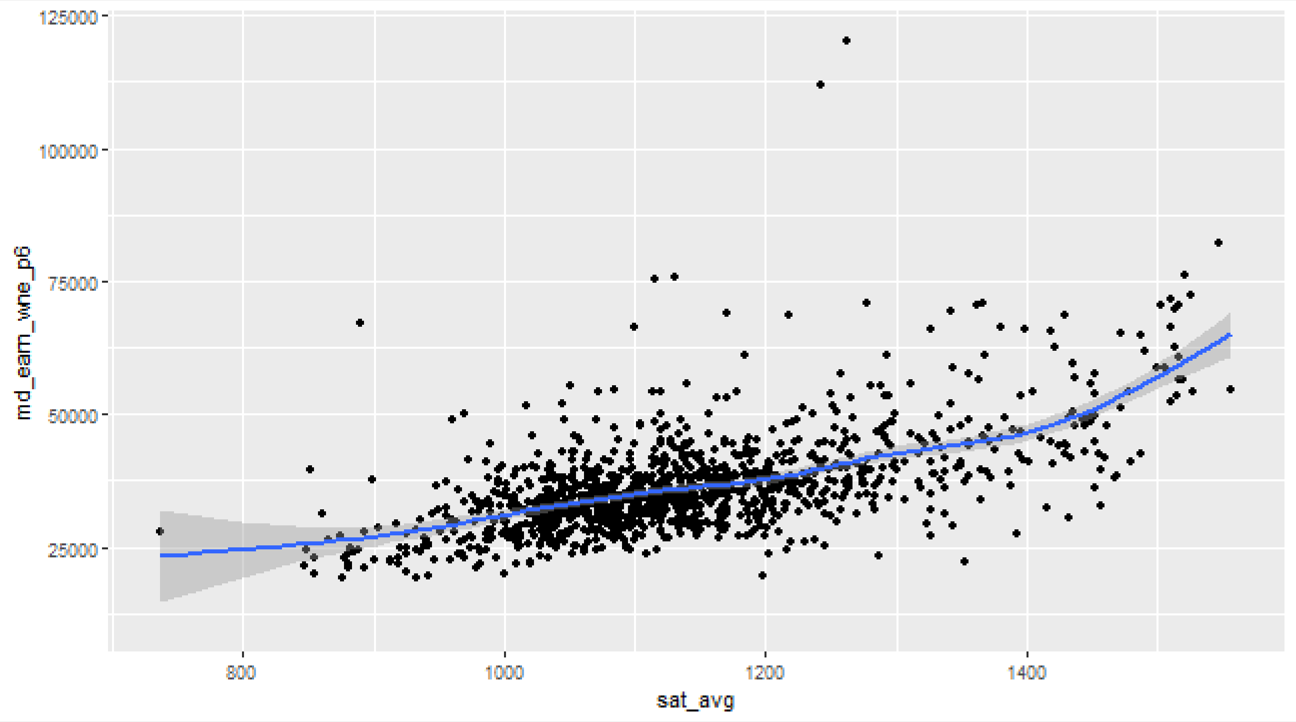
```
p + geom_text(aes(label = scales::dollar(round(mean_earn))),vjust = -.5)
```



2. Summarizing Non-parametrically

- Smoothers

```
dat %>%
  ggplot(aes(x = sat_avg,y = md_earn_wne_p6)) +
  geom_point() + geom_smooth()
```



3. Summarizing Parametrically

- Want to use *models* to describe **theoretical** relationships
- Want minimal **assumptions**...thus far?
- For inferences about μ with **large** samples
  - **identicality**: necessary for $\bar{Y}$ to be unbiased for μ
  - **independence**: necessary for $VAR(\bar{Y}) = \frac{\sigma^2}{n}$
- For inferences about μ with **small** samples
  - $Y \sim \mathcal{N}(\mu, \sigma^2)$

3. Summarizing Parametrically

- For inferences about differences in population means with **large** samples
    - Two samples are drawn **independently**
- For inferences about differences in population means with **large** samples
    - Two samples are drawn **independently**
    - Two samples have the **same variance**
    - Underlying populations are **Normal**
- This is a pretty short list!
- Lots more to come with bivariate and multivariate analysis!

3. Summarizing Parametrically

- How to describe a bivariate relationship?
- Start with notion of **correlation**

$$\begin{aligned}\rho &= \frac{COV(Y_1, Y_2)}{\sigma_1 \sigma_2} \\ &= \frac{E[(Y_1 - \mu_1)(Y_2 - \mu_2)]}{\sigma_1 \sigma_2}\end{aligned}$$

- Translating to bivariate world is easy, just use X and Y

$$\rho = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$$

- But ρ is a theoretical quantity (a **parameter**)
- What is a good estimator?

3. Summarizing Parametrically

$$r = \frac{\sum_i (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_i (X_i - \bar{X})^2 \sum_i (Y_i - \bar{Y})^2}}$$

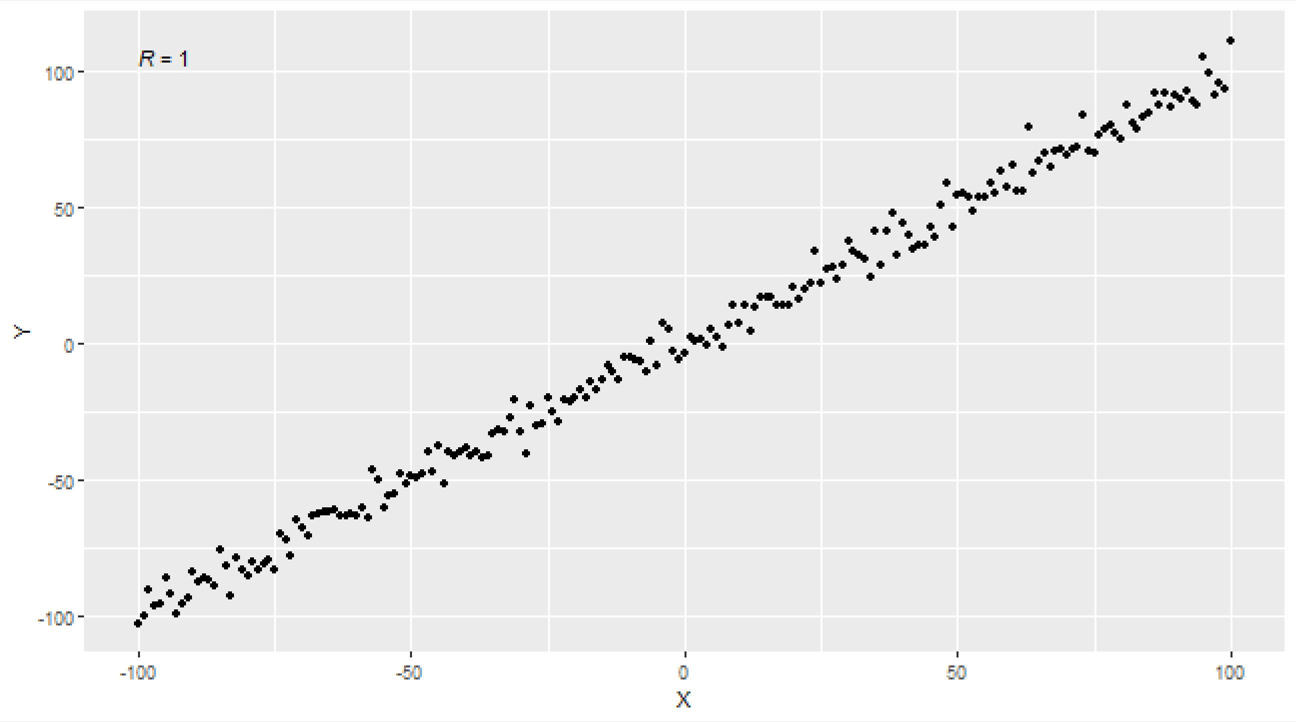
- Replace the covariance with the sample covariance $s_{XY} = \sum_i (X_i - \bar{X})(Y_i - \bar{Y})$ and the standard deviations for both variables with their sample analogues $s_X = \sqrt{\sum_i (X_i - \bar{X})^2}$ and $s_Y = \sqrt{\sum_i (Y_i - \bar{Y})^2}$.
- How good is this? It depends on the underlying data

```r
X <- seq(-100,100,by = 1)
```

3. Summarizing Parametrically

- Works well with linear relationships

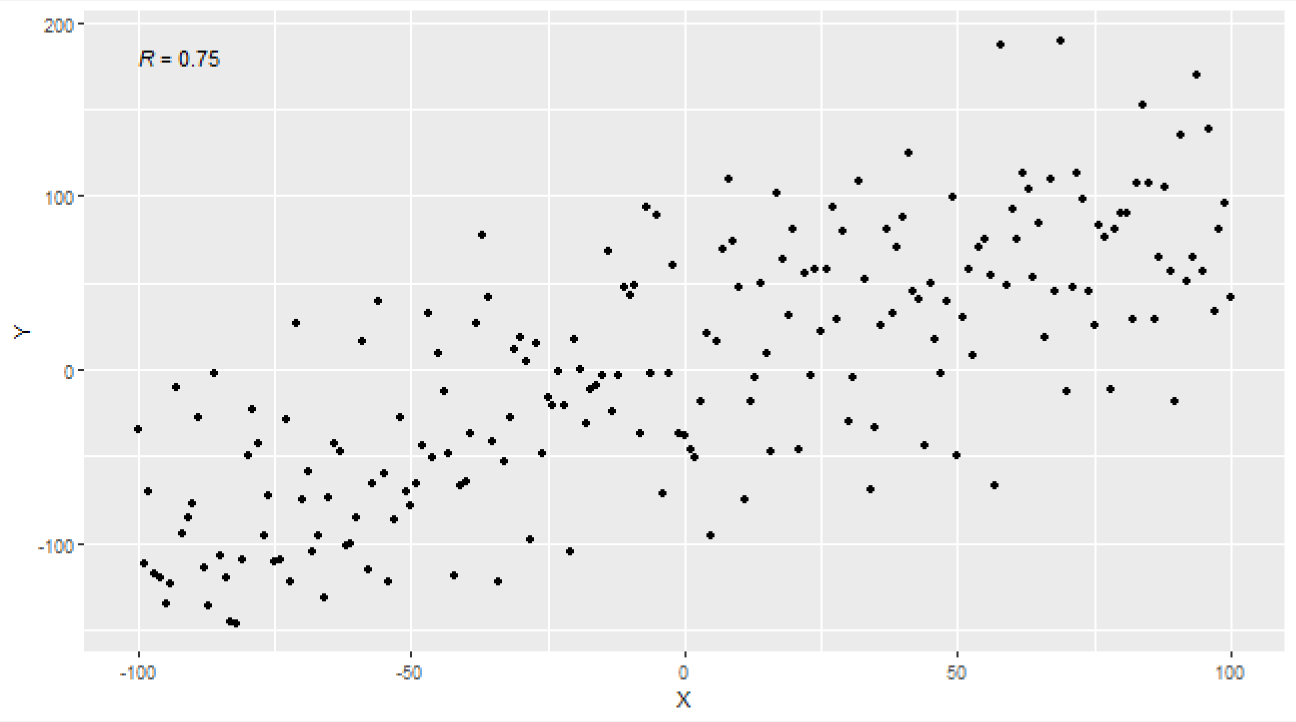
```
Y <- X + rnorm(length(X),mean = 0,sd = 5)
data.frame(X = X,Y = Y) %>%
  ggplot(aes(x = X,y = Y)) + geom_point() + stat_cor(p.digits = NA,label.sep = '')
```



3. Summarizing Parametrically

- Still works ok with more noise

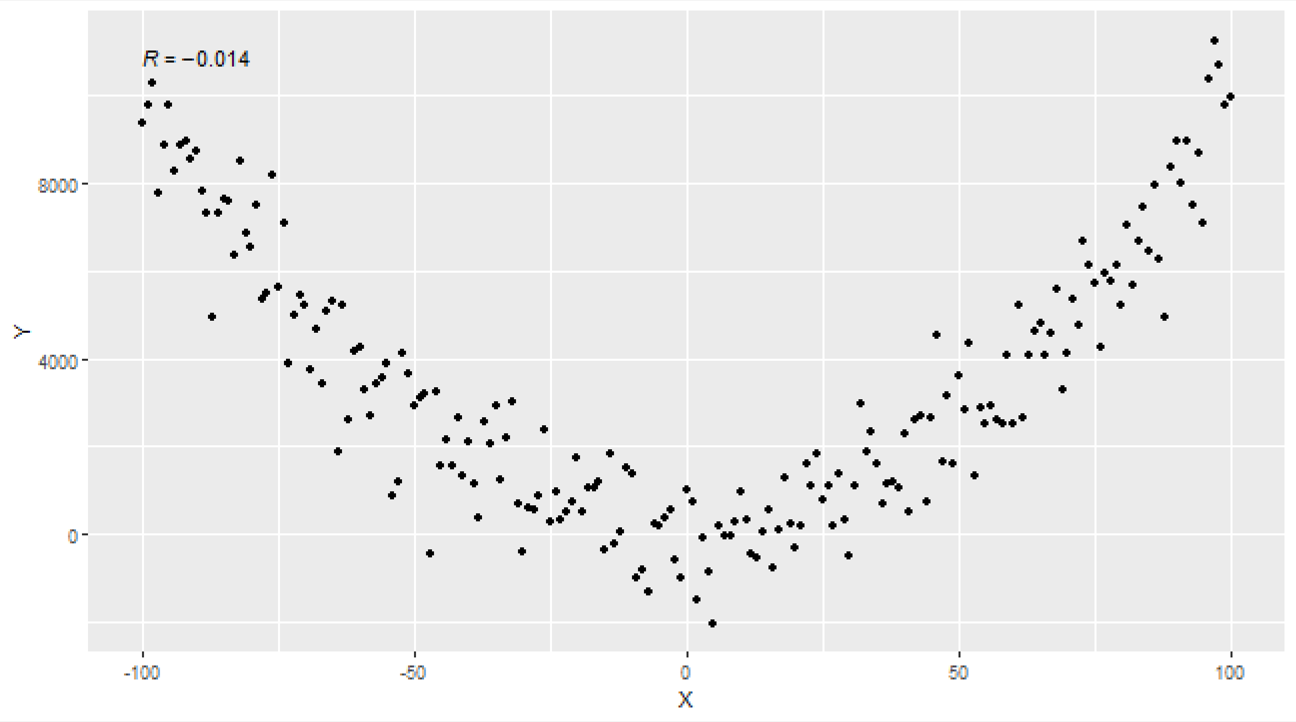
```
Y <- X + rnorm(length(X),mean = 0,sd = 50)
data.frame(X = X,Y = Y) %>%
  ggplot(aes(x = X,y = Y)) + geom_point() + stat_cor(p.digits = NA,label.sep = '')
```



3. Summarizing Parametrically

- Doesn't work well with curvelinear relationships

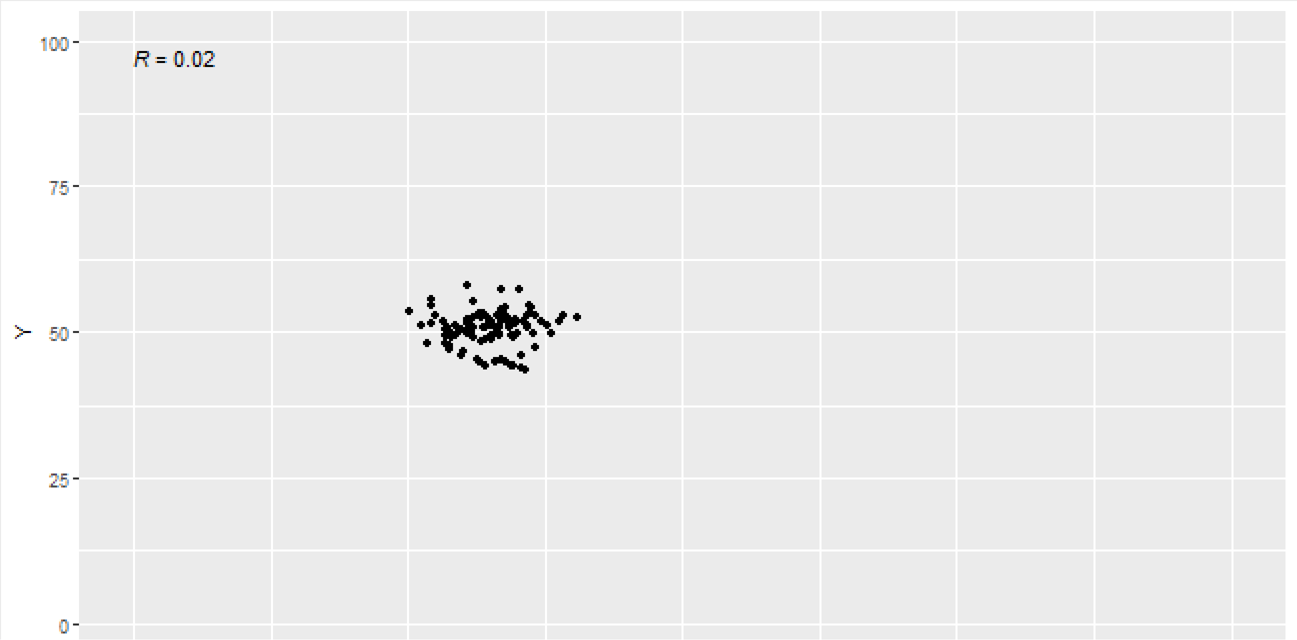
```
Y <- X^2 + rnorm(length(X),mean = 0,sd = 1000)
data.frame(X = X,Y = Y) %>%
  ggplot(aes(x = X,y = Y)) + geom_point() + stat_cor(p.digits = NA,label.sep = '')
```



3. Summarizing Parametrically

- Very sensitive to outliers

```
X <- rnorm(100,mean = 33,sd = 3)
Y <- rnorm(100,mean = 50,sd = 3)
data.frame(X = X,Y = Y) %>%
  ggplot(aes(x = X,y = Y)) + geom_point() + stat_cor(p.digits = NA,label.sep = '') + lims(x = c(0,100),y
= c(0,100))
```



3. Summarizing Parametrically

- Very sensitive to outliers

```
X[75] <- Y[75] <- 75
data.frame(X = X,Y = Y) %>%
  ggplot(aes(x = X,y = Y)) + geom_point() + stat_cor(p.digits = NA,label.sep = '') +
  xlim(c(0,100)) + ylim(c(0,100))
```