Elementary Regression Example

- This one focuses on the use of a dichotmous predictor, public or private schools
- Along the way, it runs into some data importation/coding problems.
- You can see the README here, along with the data `http://pj.freefaculty.org/guides/stat/DataSets/USNewsCollege`

Get Data For Regression

This data is provided as "raw" numbers, with no variable names.

```
if (file.exists("USNewsCollege.rds")){
   dat <- readRDS("USNewsCollege.rds")
 }else{
  dat <- read.table(url("http://pj.freefaculty.org/guides/stat/
       DataSets/USNewsCollege/USNewsCollege.csv"), header = FALSE,
       sep=",")
 mynames <- c("fice", "name", "state", "private", "avemath", "
      aveverb", "avecomb", "aveact", "fstmath", "trdmath", "fstverb"
      , "trdverb", "fstact", "trdact", "numapps", "numacc", "numenr"
      , "pctten", "pctquart", "numfull", "numpart", "instate", "
      outstate", "rmbrdcst", "roomcst", "brdcst", "addfees", "
      bookcst", "prsnl","pctphd", "pctterm", "stdtofac", "pctdonat",
       "instcst", "gradrate")
   colnames(dat) <- mynames
   saveRDS(dat, file = "USNewsCollege.rds")
 }
```

README says...

Data are from the 1995 U.S. News report on American colleges and universities. They include demographic information on tuition, room & board costs, SAT or ACT scores, application/acceptance rates, student/faculty ratio, graduation rate, and more. The dataset is used for the 1995 Data Analysis Exposition, sponsored by the Statistical Graphics Section of the American Statistical Association.

What Do We Have?

```
str(dat)
```

```
'data.frame': 1302 obs. of  35 variables:
 $ fice    : int  1061 1063 1065 11462 1002 1003 1004 1005 1009 1012
       ...
 $ name    : Factor w/ 1274 levels "Abilene Christian University",..
     : 8 994 995 993 6 327 1074 7 44 88 ...
 $ state   : Factor w/ 51 levels "AK","AL","AR",..: 1 1 1 1 2 2 2 2
     2 2 ...
 $ private : int  2 1 1 1 1 2 1 1 1 2 ...
 $ avemath : int  490 499 NA 459 NA NA NA NA 575 575 ...
 $ aveverb : int  482 462 NA 422 NA NA NA NA 501 525 ...
 $ avecomb : int  972 961 NA 881 NA NA NA NA 1076 1100 ...
 $ aveact  : int  20 22 NA 20 17 20 21 NA 24 26 ...
 $ fstmath : int  440 NA NA NA NA NA NA NA 520 470 ...
 $ trdmath : int  530 NA NA NA NA NA NA NA 638 680 ...
 $ fstverb : int  430 NA NA NA NA NA NA NA 453 460 ...
 $ trdverb : int  550 NA NA NA NA NA NA NA 559 650 ...
 $ fstact  : int  18 NA NA NA 14 NA 18 NA 21 23 ...
 $ trdact  : int  22 NA NA NA 17 NA 23 NA 27 29 ...
 $ numapps : int  193 1852 146 2065 2817 345 1351 4639 7548 805 ...
 $ numacc  : int  146 1427 117 1598 1920 320 892 3272 6791 588 ...
 $ numenr  : int  55 928 89 1162 984 179 570 1278 3070 287 ...
 $ pctten  : int  16 NA 4 NA NA NA 18 NA 25 67 ...
 $ pctquart: int  44 NA 24 NA NA 27 78 NA 57 88 ...
 $ numfull : int  249 3885 492 6209 3958 1367 2385 4051 16262 1376
```

So Many Variables, so Little Time

What does summarize report?

```
library(rockchalk)
summarize(dat)
```

```
$numerics
       addfees    aveact avecomb avemath aveverb bookcst    brdcst     fice    fstact fstmath
0%         9.0    11.000    600.0  320.00   280.0     90.0     531.0     1002    10.000  220.00
25%      130.0    20.250    884.5  460.00   422.0    480.0    1619.0     1874    18.000  410.00
50%      264.5    22.000    957.0  500.00   457.0    502.0    1980.0     2650    19.000  453.00
75%      480.0    24.000   1038.0  544.00   492.0    600.0    2402.0     3431    22.000  510.00
100%    4374.0    31.000   1410.0  750.00   665.0   2340.0    6250.0    30430    29.000  740.00
mean     392.0    22.120    968.0  506.80   461.2    550.0    2061.0     3126    19.820  462.20
sd       469.4     2.580    123.6   67.82    58.3    167.4     661.7     2970     2.796   76.32
var   220300.0     6.656  15270.0 4600.00  3399.0 28010.0 437900.0  8822000     7.819 5825.00
NA's     274.0   588.000    523.0  525.00   525.0     48.0     498.0        0   639.000  530.00
N       1302.0  1302.000   1302.0 1302.00  1302.0   1302.0    1302.0     1302  1302.000 1302.00
       fstverb gradrate  instate  instcst     numacc    numapps   numenr  numfull    numpart
0%      200.00     8.00      480     1834       35.0  3.500e+01     18.0       59        1.0
25%     380.00    47.00     2580     6116      554.5  6.958e+02    236.0      966      131.2
50%     410.00    60.00     8050     7729     1095.0  1.470e+03    447.0     1812      472.0
75%     450.00    74.00    11600    10050     2303.0  3.314e+03    984.0     4540     1313.0
100%    630.00   118.00    25750    62470    26330.0  4.809e+04   7425.0    31640    21840.0
mean    418.50    60.41     7897     8988     1871.0  2.752e+03    778.9     3693     1082.0
sd       64.49    18.89     5348     5347     2251.0  3.542e+03    884.6     4545     1672.0
var    4159.00   356.80 28600000 28600000  5066000.0  1.255e+07 782500.0 20660000 2796000.0
NA's    530.00    98.00       30       39       11.0  1.000e+01      5.0        3       32.0
N      1302.00  1302.00     1302     1302     1302.0  1.302e+03   1302.0     1302     1302.0
       outstate pctdonat  pctphd pctquart  pctten pctterm   private    prsnl rmbrdcst
0%         1044     0.00     8.00     6.00    1.00   20.00    1.0000     75.0      1260
25%        6111    11.00    57.00    36.75   13.00   63.00    1.0000    900.0      3320
```

So Many Variables, so Little Time ...

```
50%      8670    19.00   71.00    50.00   21.00   77.00    2.0000   1250.0     4030
75%     11660    29.00   82.00    66.00   32.00   90.00    2.0000   1794.0     4849
100%    25750    81.00  105.00   100.00   98.00  100.00    2.0000   6900.0     8700
mean     9277    20.91   68.65    52.35   25.67   75.23    1.6390   1389.0     4162
sd       4171    12.67   17.83    20.88   18.31   17.11    0.4805    714.2     1179
var  17400000   160.60  317.80   436.00  335.40  292.70    0.2309 510200.0  1391000
NA's       20   222.00   32.00   202.00  235.00   30.00    0.0000    181.0       76
N        1302  1302.00 1302.00  1302.00 1302.00 1302.00 1302.0000   1302.0     1302
     roomcst stdtofac   trdact trdmath trdverb
0%       500    2.300   15.000  330.00  330.00
25%     1710   11.800   23.000  530.00  480.00
50%     2200   14.300   25.000  580.00  530.00
75%     3040   17.600   27.000  630.00  570.00
100%    7400   91.800   35.000  785.00  720.00
mean    2515   14.860   25.110  583.10  530.50
sd      1151    5.186    2.781   71.22   64.54
var  1324000   26.900    7.735 5072.00 4165.00
NA's     321    2.000  639.000  530.00  530.00
N       1302 1302.000 1302.000 1302.00 1302.00

$factors
                 name                        state
 Bethel College    :   4.0000   NY             : 101.0000
 Concordia College:   4.0000   PA             :  83.0000
 Trinity College   :   4.0000   CA             :  70.0000
 Columbia College  :   3.0000   TX             :  60.0000
 (All Others)      :1287.0000   (All Others)   : 988.0000
 NA's              :   0.0000   NA's           :   0.0000
 entropy           :  10.2977   entropy        :   5.2502
 normedEntropy     :   0.9983   normedEntropy  :   0.9256
 N                 :1302.0000   N              :1302.0000
```

Create a factor variable "schtype"

- Recode private from numeric 1-2 to a new factor variable called "schtype":

```
dat$schtype <- factor(dat$private, levels = c("1","2"), labels = c(
    "public","private"))
## Proofread
table(dat$schtype, dat$private, exclude = NULL)
```

```
           1   2 <NA>
 public  470   0    0
 private   0 832    0
 <NA>      0   0    0
```

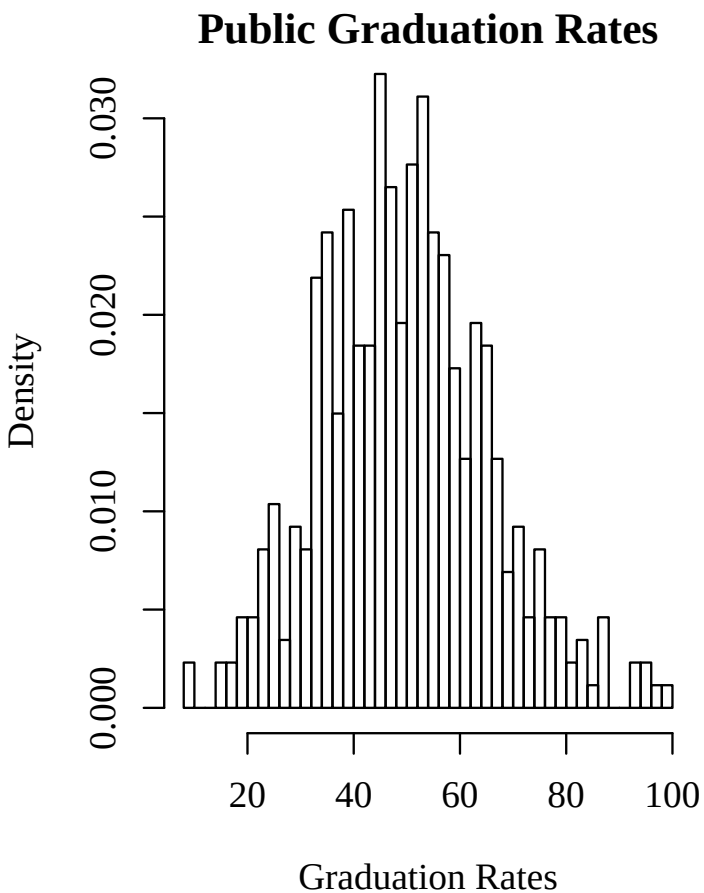
```
levels(dat$schtype)
```

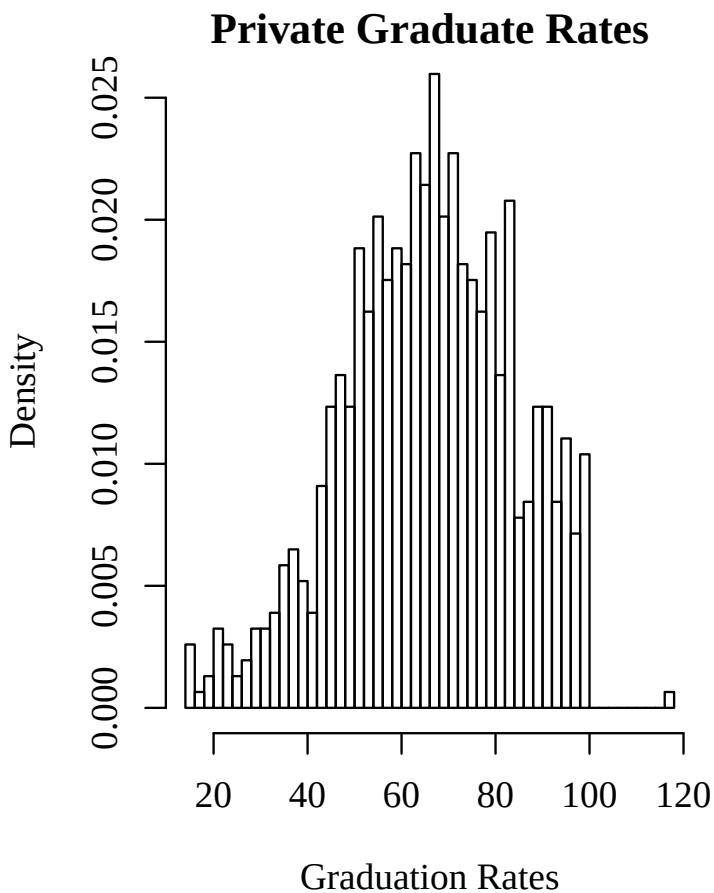
```
[1] "public"  "private"
```

- Keep the numeric variables gradrate, instcst, stdtofac, aveact, instate, outstate

```
dat <- dat[, c("gradrate", "instcst", "stdtofac", "aveact", "
    instate", "outstate", "schtype")]
```

Subset Public and Private For Comparison





Subset Public and Private For Comparison

- How did he do that?

```
datpublic  <- dat[dat$schtype %in% levels(dat$schtype)[1], ]
datprivate <- dat[dat$schtype %in% levels(dat$schtype)[2], ]
par(mfcol = c(1,2))
hist(datpublic$gradrate, prob = TRUE, breaks = 50, main = "Public
    Graduation Rates", xlab = "Graduation Rates")
hist(datprivate$gradrate, prob = TRUE, breaks = 50, main = "Private
     Graduate Rates", xlab = "Graduation Rates")
```

- Note levels(dat$schtype)[1] == "public". I prefer to choose by levels(dat$schtype)[1], rather than "private", seems less typo-prone
- I explicitly create subsets, and named separately for clarity. You'll find a lot of different ways to do this
  -

```
lapply(levels(dat$schtype), function(x) hist(dat$gradrate[dat$
    schtype==x], prob = TRUE, breaks = 50, main= paste(x, "
    Graduation Rates"), xlab = "Graduation Rates"))
```

Subset Public and Private For Comparison ...

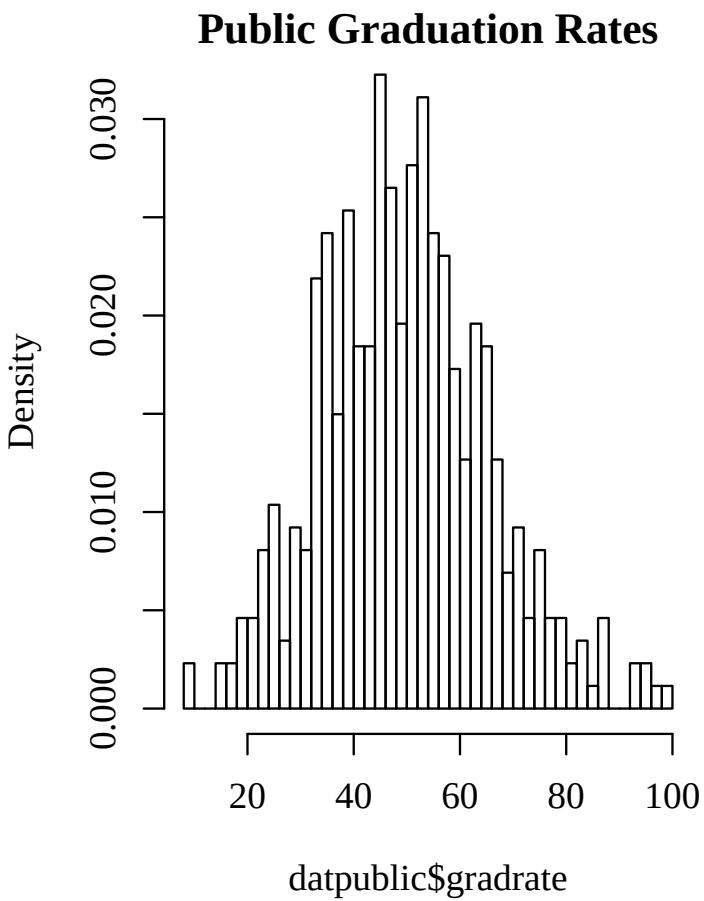
```
hist(dat[dat$schtype==levels(dat$schtype)[1], "gradrate"], prob
    = TRUE, breaks = 50, main="Public Graduation Rates", xlab
   = "Graduation Rates")
```

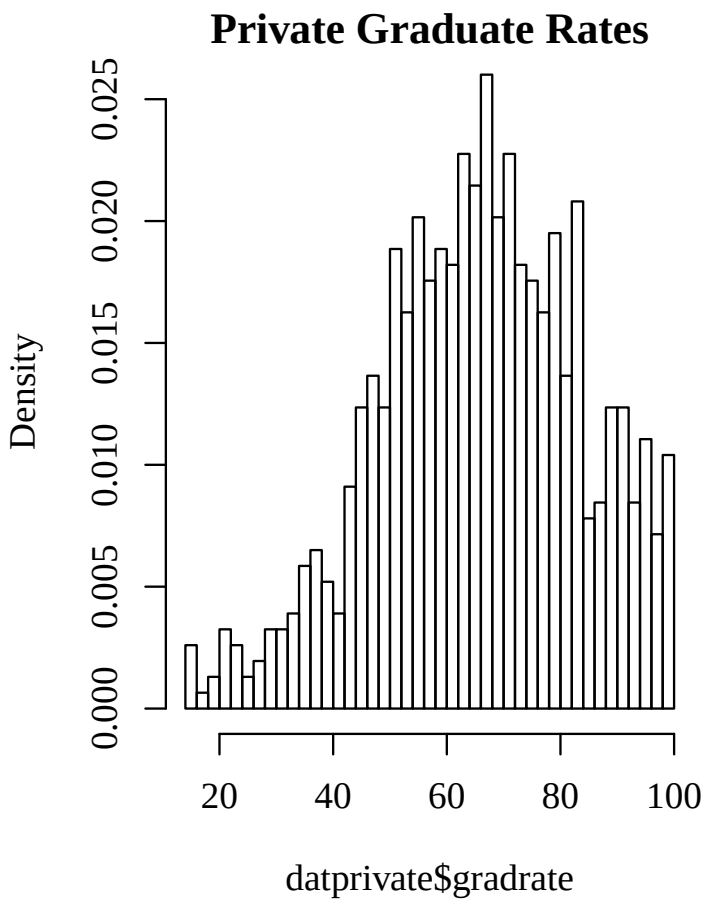
Should we correct the data for private schools?

Obvious data entry error: Some graduation rates are over 100.
Lets trim them out of the data, and re-create the public/private subsets

```
dat <- dat[ dat$gradrate ≤ 100, ]
datpublic  <- dat[dat$schtype %in% levels(dat$schtype)[1], ]
datprivate <- dat[dat$schtype %in% levels(dat$schtype)[2], ]
par(mfcol=c(1,2))
hist(datpublic$gradrate, prob = TRUE, breaks = 50, main = "Public
    Graduation Rates")
hist(datprivate$gradrate, prob = TRUE, breaks = 50, main = "Private
     Graduate Rates")
```

Should we correct the data for private schools? ...





Grab Some Summary Stats I Might Need Later

summary(dat) would be a nice start, but the output hard to manage. So Build own summary:

```
summarize(datpublic)
```

```
$numerics
       aveact  gradrate  instate   instcst outstate stdtofac
0%     16.000      8.00      480      1834     1044     6.700
25%    20.000     39.00     1681      5265     4738    15.120
50%    21.000     50.00     2100      6414     6114    17.600
75%    23.000     59.75     2679      8046     7439    19.600
100%   28.000    100.00    14320     46480    15730    29.500
mean   21.460     50.18     2231      7320     6226    17.440
sd      2.114     15.81     1064      4170     2162     3.525
var     4.470    250.10  1132000  17390000  4672000    12.420
NA's  197.000      0.00       22         8       12     0.000
N     434.000    434.00      434       434      434   434.000

$factors
            schtype
 public         :434
 NA's           :  0
 entropy        :  0
 normedEntropy:  0
```

Grab Some Summary Stats I Might Need Later ...

```
N                :434
```

```
summarize(datprivate)
```

```
$numerics
       aveact  gradrate   instate   instcst outstate stdtofac
0%     11.000     15.00      1254      3019     2340      2.50
25%    21.000     54.00      8564      7051     8571     11.10
50%    22.000     67.00     10800      8694    10800     12.80
75%    24.000     79.00     13590     11220    13590     14.70
100%   31.000    100.00     25750     62470    25750     91.80
mean   22.700     66.10     11250     10160    11270     13.25
sd      2.675     17.95      4011      5819     3967      4.72
var     7.157    322.00  16090000  33870000 15740000     22.27
NA's  347.000      0.00         6        17        5      0.00
N     769.000    769.00       769       769      769    769.00

$factors
             schtype
 private         :769
 NA's            :  0
 entropy         :  0
 normedEntropy:  0
 N               :769
```

Grab Some Summary Stats I Might Need Later ...

Describe the variable schtype

```
(t1 <- table(dat$schtype))
```

```
public private
   434     769
```

```
prop.table(t1)
```

```
   public    private
0.3607648 0.6392352
```

Conduct T-test on Graduation Rate Difference

- The default t.test in R will use "Welch corrected" estimate of the standard error

```
t.test(gradrate ~ schtype, data = dat)
```

```
  Welch Two Sample t-test

data:  gradrate by schtype
t = -15.9578, df = 994.788, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -17.87557 -13.96064
sample estimates:
 mean in group public mean in group private
             50.18203              66.10013
```

- But it used to assume the variances in the 2 groups are equal

```
t.test(gradrate ~ schtype, data = dat, var.equal = TRUE)
```

Conduct T-test on Graduation Rate Difference ...

```
  Two Sample t-test

data:  gradrate by schtype
t = -15.4082, df = 1201, p-value < 2.2e-16
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -17.94497 -13.89123
sample estimates:
 mean in group public mean in group private
             50.18203              66.10013
```

- Keep an eye on this latter set of estimates, compare to the lm output.

Regress Graduate Rates on the Dichotomy "schtype"

```
mod1 <- lm(gradrate ~ schtype, data = dat)
summary(mod1)
```

```
Call:
lm(formula = gradrate ~ schtype, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-51.10 -11.18  -0.10   11.90   49.82

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)    50.182      0.826   60.75   <2e-16 ***
schtypeprivate 15.918      1.033   15.41   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 17.21 on 1201 degrees of freedom
  (98 observations deleted due to missingness)
Multiple R^2:  0.1651,  Adjusted R^2:  0.1644
F-statistic: 237.4 on 1 and 1201 DF,  p-value: < 2.2e-16
```

Check "model.matrix" To See What R Did to "schtype"

```
head(model.matrix(mod1))
```

```
  (Intercept) schtypeprivate
1           1               1
3           1               0
5           1               0
6           1               1
7           1               0
8           1               0
```

Regression Table

	M1 Estimate (S.E.)
(Intercept)	50.182*** (0.826)
schtypeprivate	15.918*** (1.033)
N	1203
RMSE	17.207
R^2	0.165
$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$	

- Estimated Intercept
- Estimated Slope
- Standard Error of Intercept Estimate (estimated standard deviation of intercept estimator)
- Standard Error of Slope Estimate (estimated standard deviation of slope estimator)

Hypothesis Test for Slope

- Theory: $gradrate_i = c_0 + c_1$"*contrast for private*" $+ u_i$
  c_0 and c_1 are real-valued constants, $E[u_i] = 0$, $Var[u_i] = E[u_i^2] = \sigma_u^2$.
- The Null Hypothesis: $H_0 : c_1 = 0$
- $\hat{c}_1$ is approximately Normal, So create T test:
- The critical value of t is:

```
> qt( c(0.025, 0.975), df=90)
[1] -1.986675  1.986675
```

- Conclusion: "The estimate $\hat{c}_1$ is statistically significantly different from 0."

Confidence Intervals for Intercept and Slope

```
confint(mod1)
```

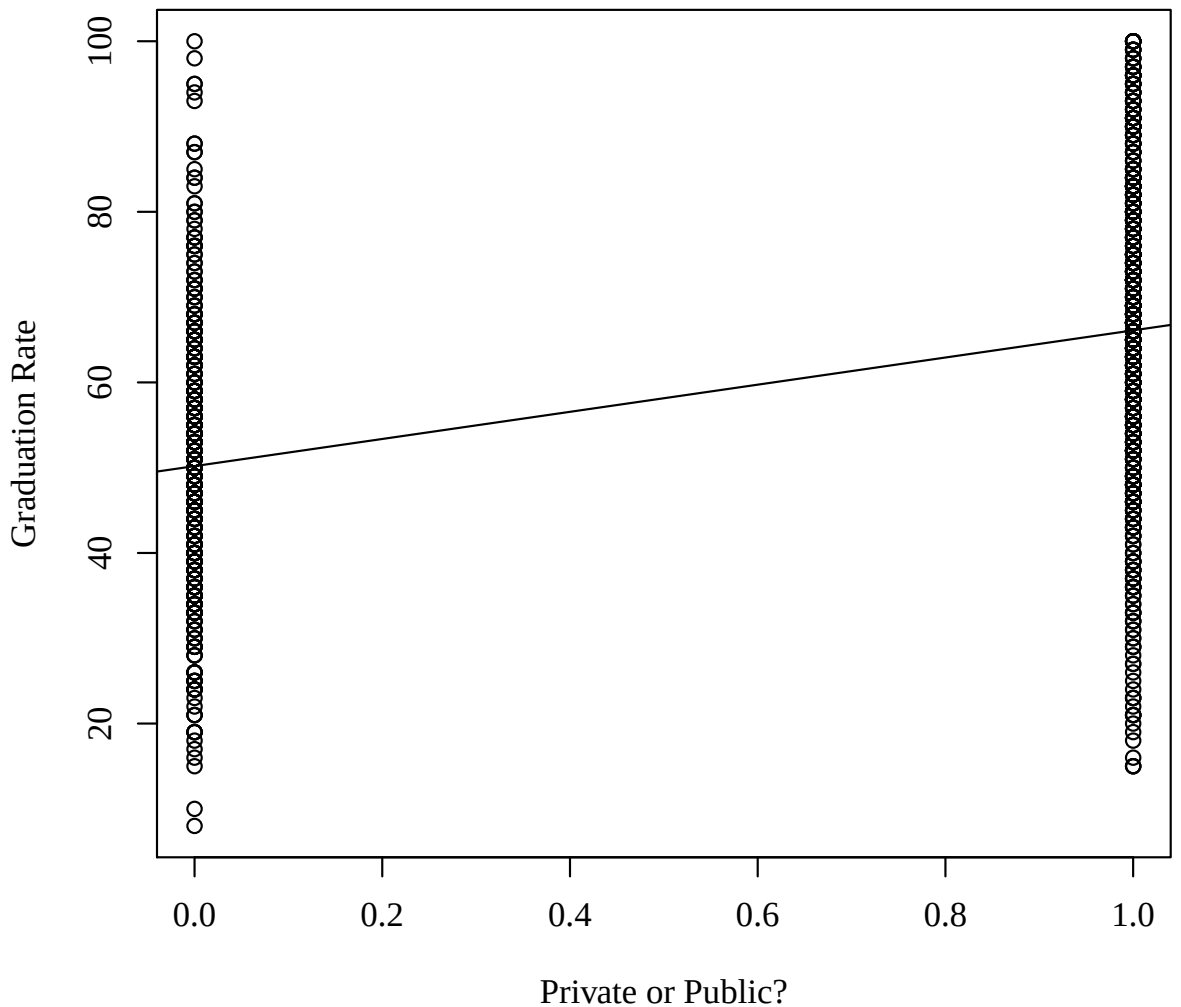
```
                   2.5 %   97.5 %
(Intercept)     48.56150 51.80256
schtypeprivate 13.89123 17.94497
```

Supposing the model's theory is correct, we believe

- the estimated slope $\hat{c}_1$ would be between 0.0079 and 0.125 in 95% of repeated samples from same process
- the probability that the true slope c_1 is in that range is 0.95.

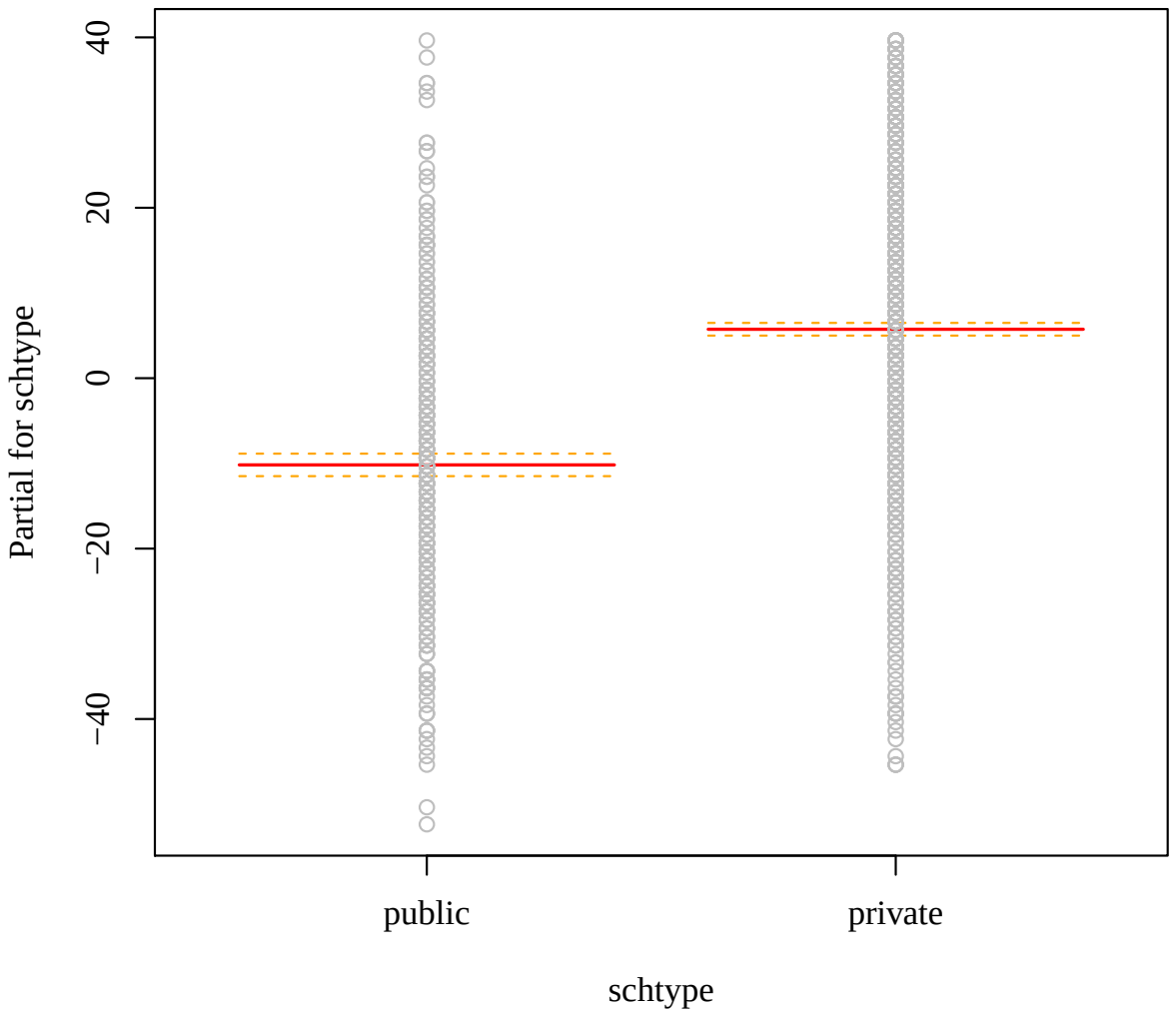
Draw Predicted Values

We can force R to plot schtype as if it were a numeric variable



But is that a meaningful line?

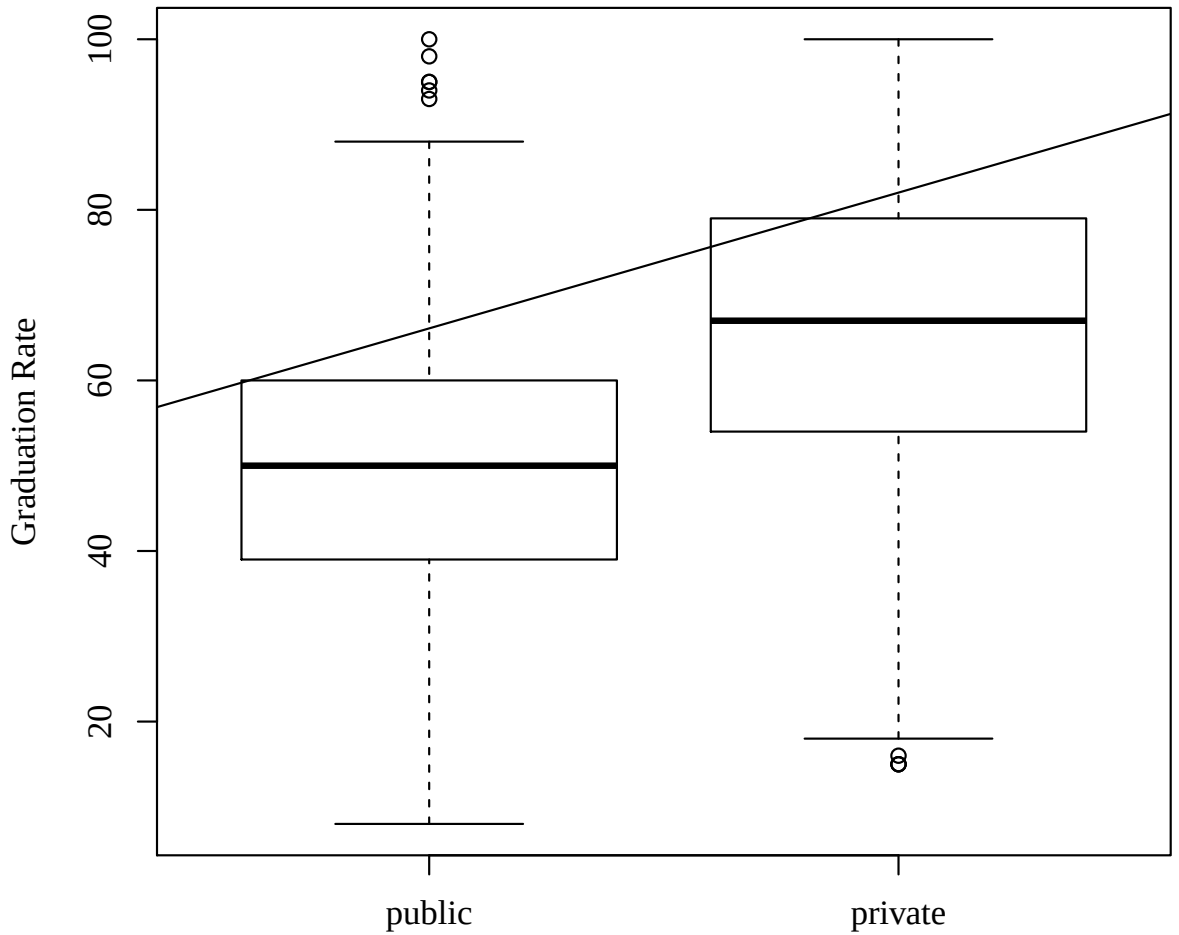
The termplot Draws Predicted Values




```
termplot(mod1, terms = c("schtype"), partial = TRUE, se = TRUE)
```

Perhaps a Box Plot is a Better Way to Look at this

plot notices "schtype" is a factor and it creates a boxplot.
But "abline(mod1)" draws a line in the wrong place



A word about Predicted Values

With only two "levels" of private, there are only two unique predicted values. The predict function will run, and apply to each row of data, but it just generates the same thing over and over.
Run this to see whats going on.

```
bp1 <- predict(mod1, interval = "confidence")
head(bp1)
```

```
       fit      lwr      upr
1 66.10013 64.88272 67.31754
3 50.18203 48.56150 51.80256
5 50.18203 48.56150 51.80256
6 66.10013 64.88272 67.31754
7 50.18203 48.56150 51.80256
8 50.18203 48.56150 51.80256
```

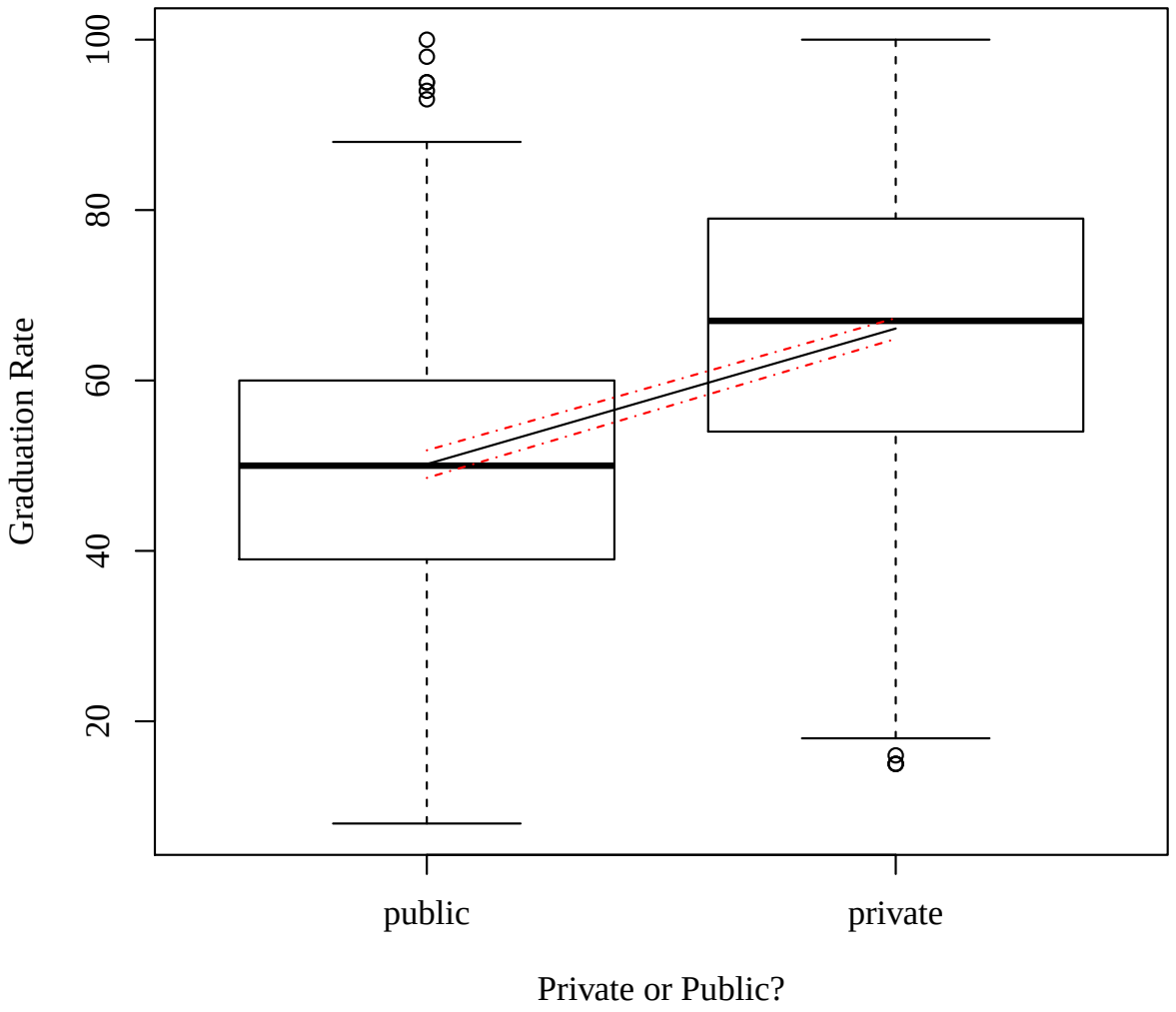
A Word about Predicted Values

So we could just create a new small data set and get predictions for it.

```
newdf <- data.frame(schtype = factor(c("public", "private"), levels
      = c("public", "private")))
gradratehat <- predict(mod1, interval = "conf", newdata = newdf)
newdf <- cbind(gradratehat, newdf)
newdf
```

```
       fit      lwr      upr schtype
1 50.18203 48.56150 51.80256  public
2 66.10013 64.88272 67.31754 private
```

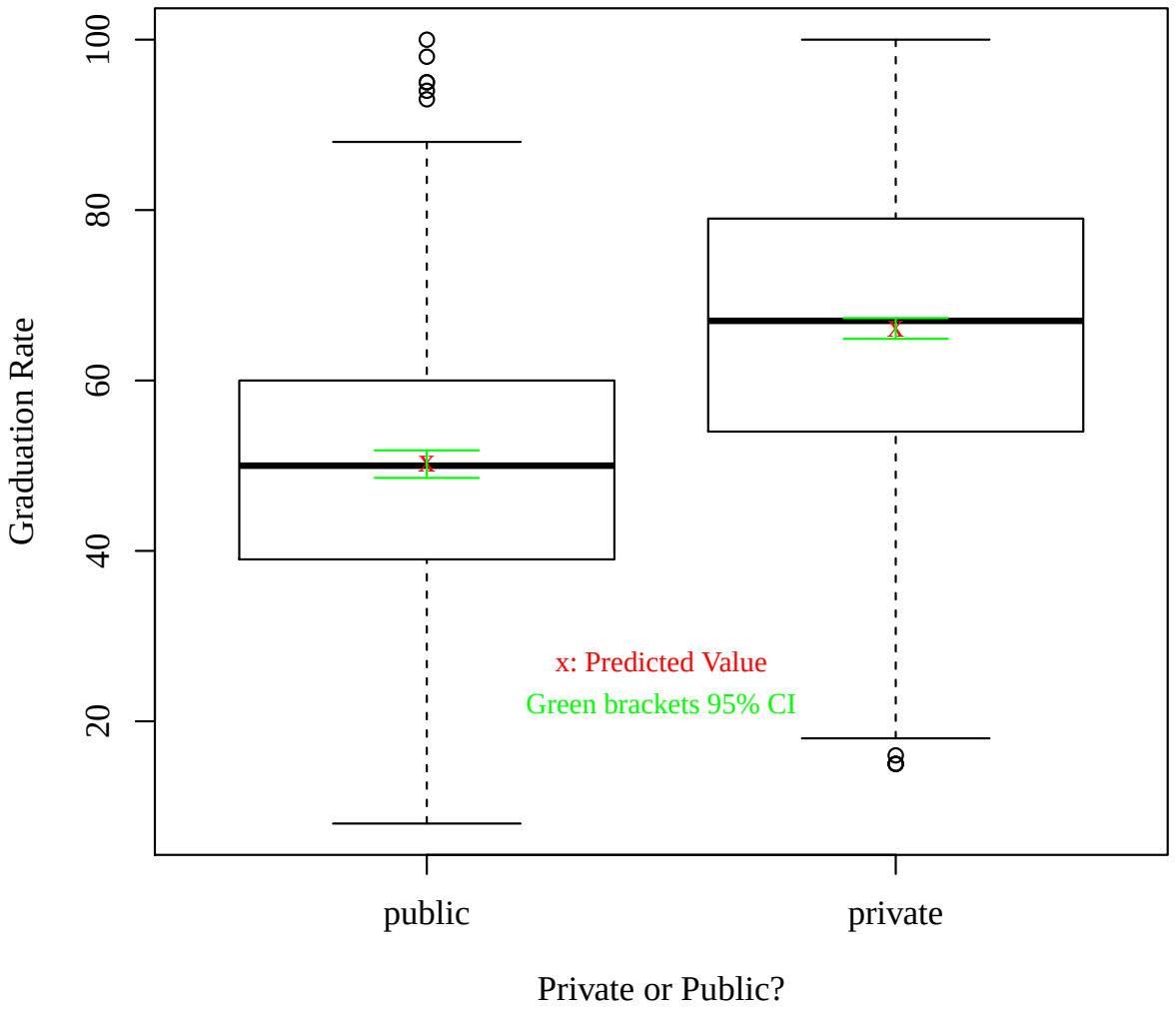
But The Predicted Value Lines Still Look Stupid, IMHO



But The Predicted Value Lines Still Look Stupid, IMHO ...

```
#newdf <- data.frame(schtype = levels(dat$schtype))
#newdf$schtype <- relevel(newdf$schtype, "public") ###WOW:  problem
      required this
plot(gradrate ~ schtype, data = dat, xlab = "Private or Public?",
     ylab = "Graduation Rate", ylim = range(dat$gradrate, na.rm =
     TRUE))
gradratehat <- predict(mod1, interval="conf", newdata=newdf)
lines(gradratehat[ , 1] ~ schtype, data = newdf, col = "black", lty
      = 1)
lines(gradratehat[ , 2] ~ schtype, data = newdf, col = "red", lty =
      4)
lines(gradratehat[ , 3] ~ schtype, data = newdf, col = "red", lty =
      4)
```

Maybe A boxplot with the Predicted Values Marked In?



Maybe A boxplot with the Predicted Values Marked In? ...

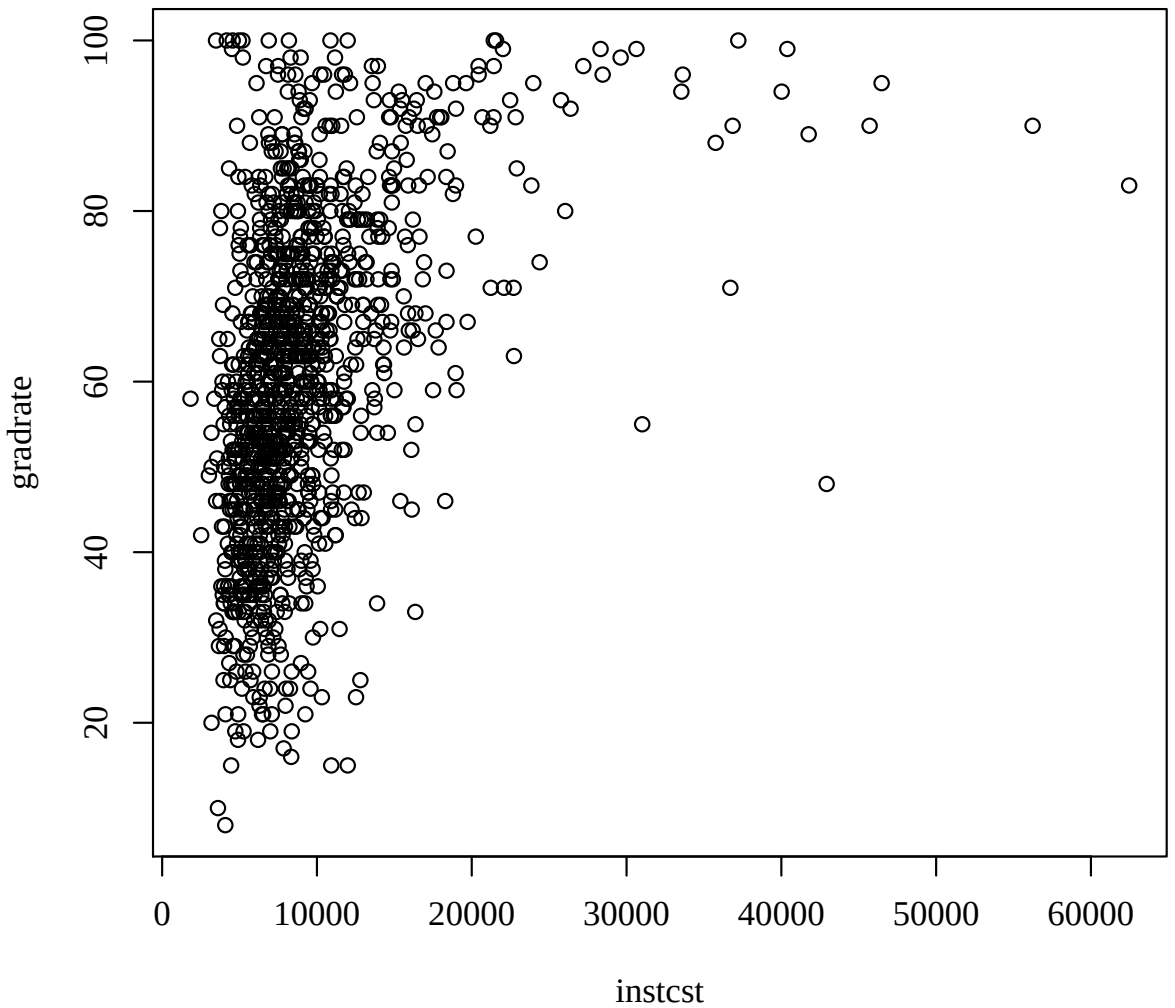
Boxes center on the median, not the mean (predicted values), thus a little mismatch.
This whole exercise is just making me feel stupider and stupider

```
plot(gradrate ~ schtype, data = dat, xlab = "Private or Public?",
     ylab = "Graduation Rate", ylim = range(dat$gradrate, na.rm =
     TRUE))
points(gradratehat[ , 1] ~ as.numeric(schtype), data = newdf, pch =
      "x", col = "red", cex = 1)
arrows(x0 = as.numeric(newdf$schtype), y0 = gradratehat[ , 2],x1=
     as.numeric(newdf$schtype), y1 = gradratehat[ , 3], col = "
     green", angle = 90, code = 3)
text( c(30,25) ~ c( 1.5, 1.5), pos = 1, adj = 2, label = c("x:
     Predicted Value", "Green brackets 95% CI"), cex = 0.8, col = c(
     "red", "green"))
```

That Considered Just One Categorical Predictor

- This lecture is about one-predictor regressions, so I can't get too fancy
- But I'd like to explore the idea that we might find a different relationship between 2 variables when we compare the public and private school datasets
- When we later turn to multiple regression, we'll see different==better ways to do this, but this is not a bad start.

Consider Instructional Spending




```
plot(gradrate ~ instcst, data=dat)
```

Consider Instructional Spending as a Predictor of gradrate

```
mod2 <- lm(gradrate ~ instcst, data = dat)
summary(mod2)
```

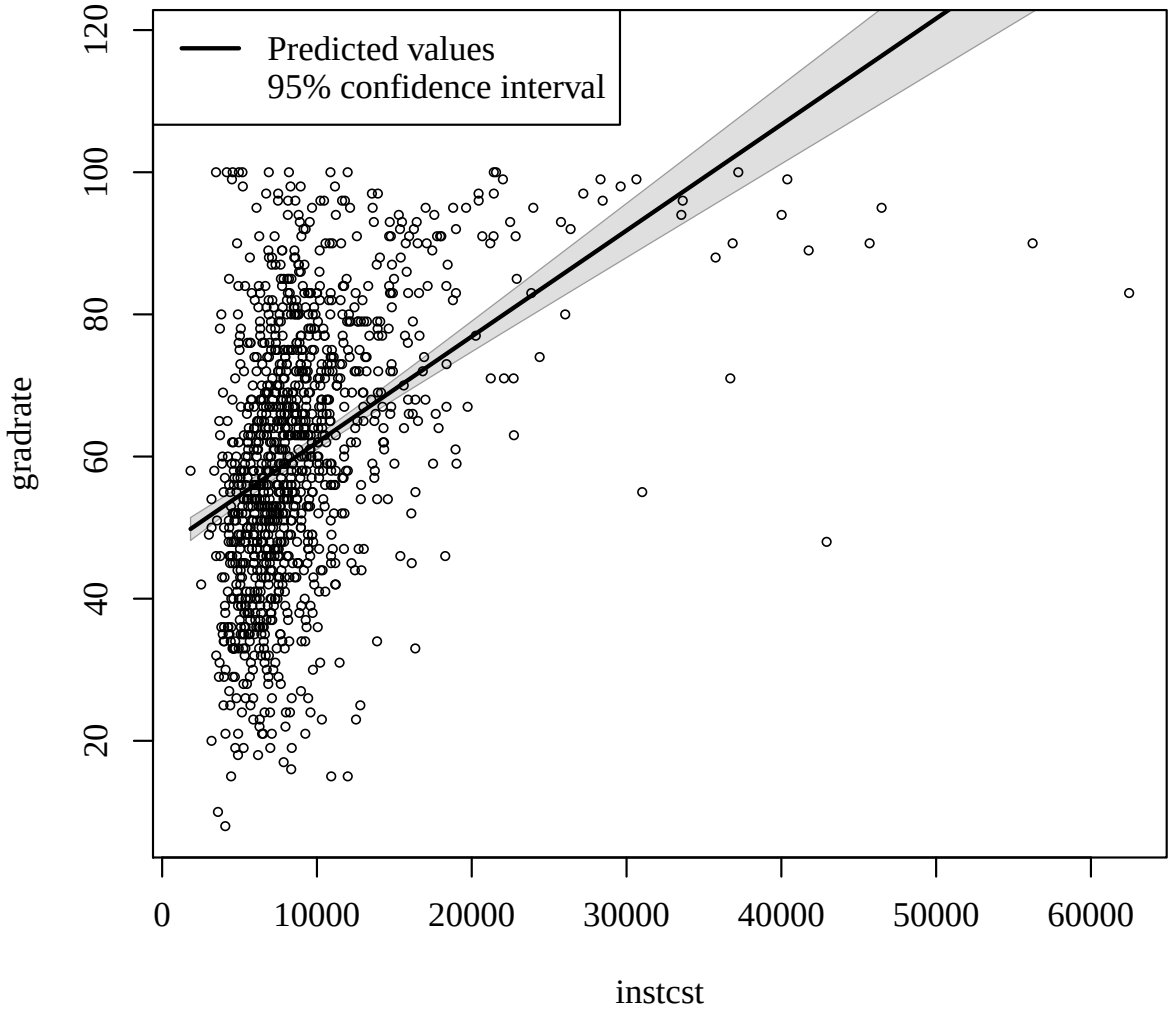
```
Call:
lm(formula = gradrate ~ instcst, data = dat)

Residuals:
    Min      1Q  Median      3Q     Max
-63.072 -10.687  -0.023  11.442  47.739

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.707e+01  9.577e-01   49.15   <2e-16 ***
instcst     1.491e-03  9.003e-05   16.56   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.85 on 1176 degrees of freedom
  (123 observations deleted due to missingness)
Multiple R^2:  0.1891,  Adjusted R^2:  0.1884
F-statistic: 274.3 on 1 and 1176 DF,  p-value: < 2.2e-16
```

Problem 1 Obvious



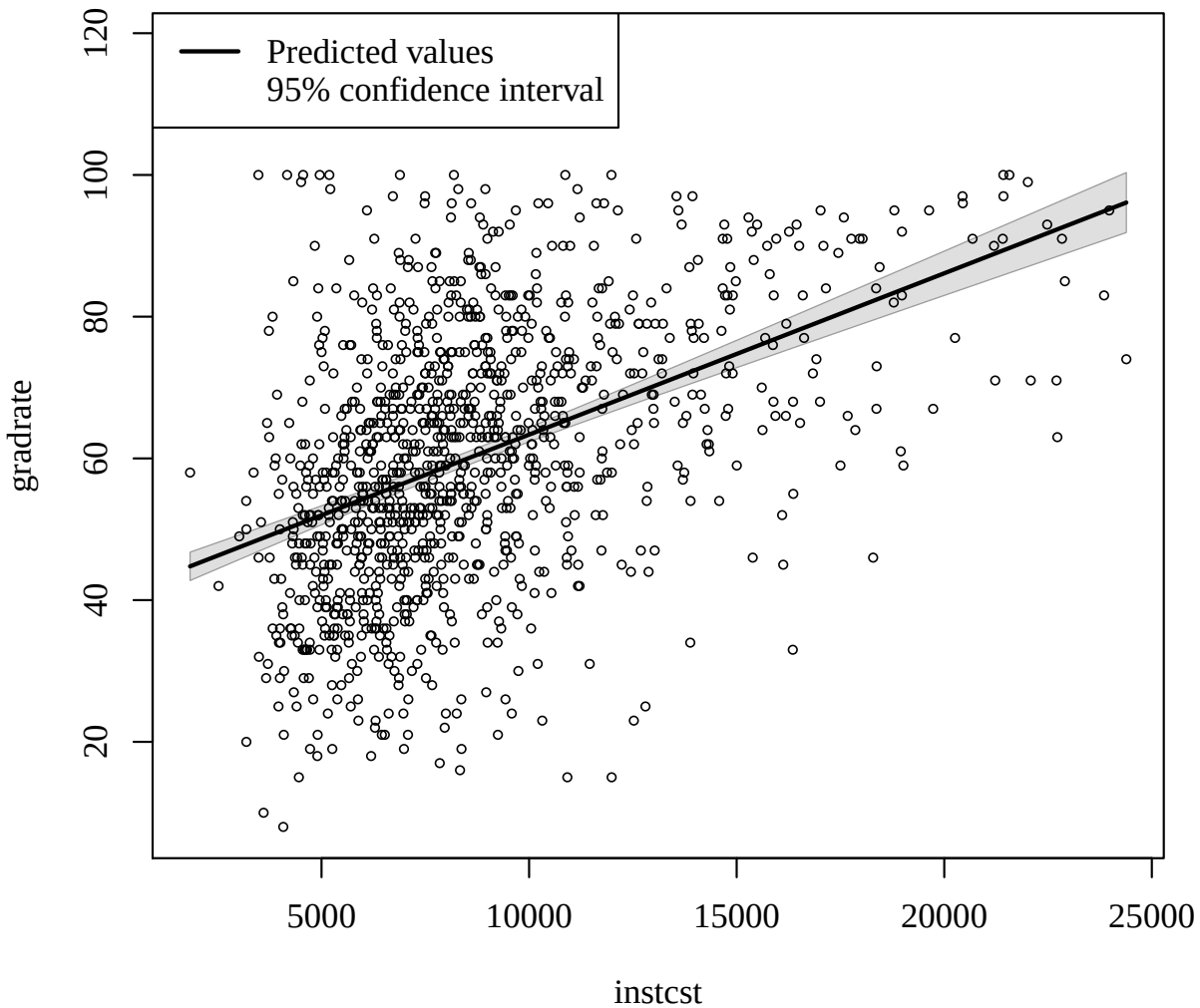

```
plotSlopes(mod2, plotx = "instcst", interval = "confidence")
```

A Few Big Spenders Just Don't Fit

- In a real study, I wouldn't throw out those cases without a lot more checking
- I'd consider making a model that "bends" through the data
- Right now, however, lets just focus on the schools for which the data has instcst < 25000.

```
mod3 <- lm(gradrate ~ instcst, data = dat, subset = instcst <
    25000)
plotSlopes(mod3, plotx = "instcst", interval = "confidence")
```

Fitted when Instructional Spending $<$ \$25,000



Regression Table

	M1 Estimate (S.E.)
(Intercept)	40.591***
	(1.240)
instcst	0.002***
	(0.000)
N	1155
RMSE	16.410
R^2	0.203

$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$

Identify

- Estimated Intercept & Slope
- Estimated Standard Errors
- Estimate of error term's standard deviation)

Rescale instcst to 1000s of dollars

- Because the instcst variable is in dollars, the coefficients are very small
- To make them more readable, lets rescale instcst
- From now on, instcstP1000 = instcst / 1000

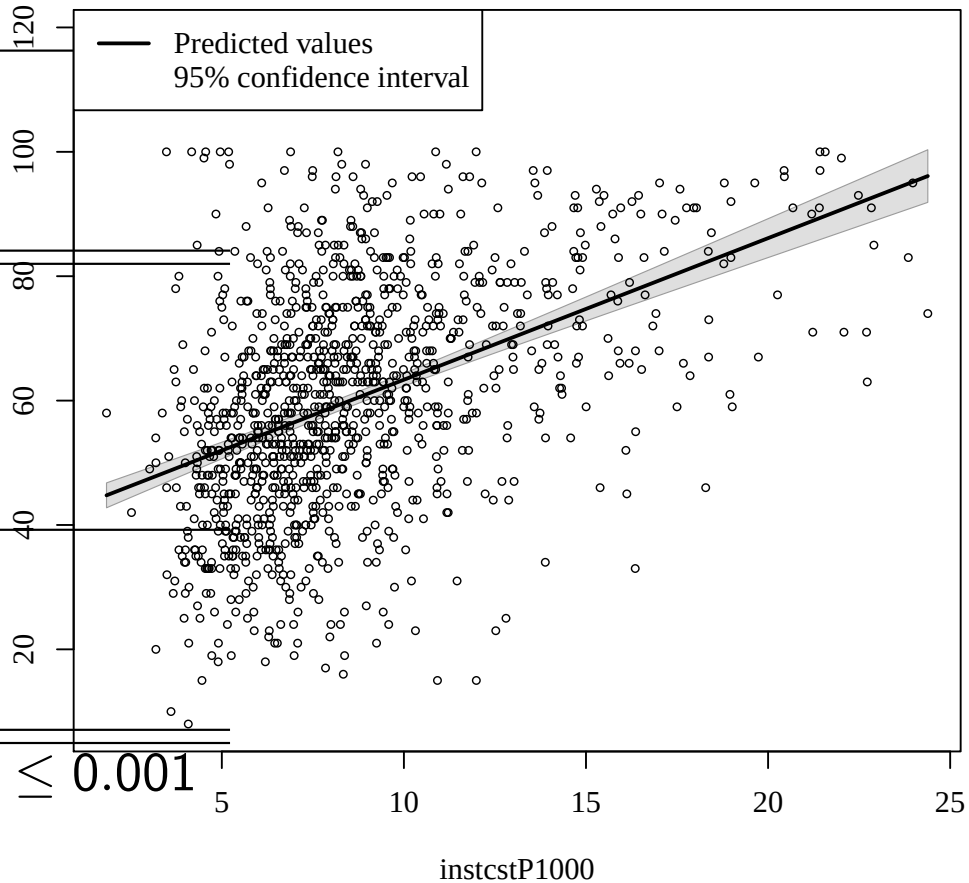
Fitted Model for Instructional Spending

```
dat$instcstP1000 <- dat$instcst/1000
mod3 <- lm(gradrate ~ instcstP1000, data = dat, subset =
     instcstP1000 < 25)
```

Results with instcstP1000

	M1
	Estimate
	(S.E.)
(Intercept)	40.591***
	(1.240)
instcstP1000	2.276***
	(0.133)
N	1155
RMSE	16.410
R^2	0.203

$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$



Hypothesis Test for Slope

- Theory: $gradrate_i = \beta_0 + \beta_1 instcst + e_i$
  β_0 and β_1 are real-valued constants, $E[e_i] = 0$, $Var[e_i] = E[e_i^2] = \sigma_e^2$.
- The Null Hypothesis: $H_0 : \beta_1 = 0$
- $\hat{\beta}_1$ is approximately Normal, So create T test:
- The critical value of t is:

```
> qt( c(0.025, 0.975), df=90)
[1] -1.986675  1.986675
```

- Conclusion: "The estimate $\hat{\beta}_1$ is statistically significantly different from 0."

Confidence Intervals for Intercept and Slope

```
confint(mod3)
```

```
                 2.5 %    97.5 %
(Intercept)  38.159150 43.023565
instcstP1000  2.015515  2.537295
```

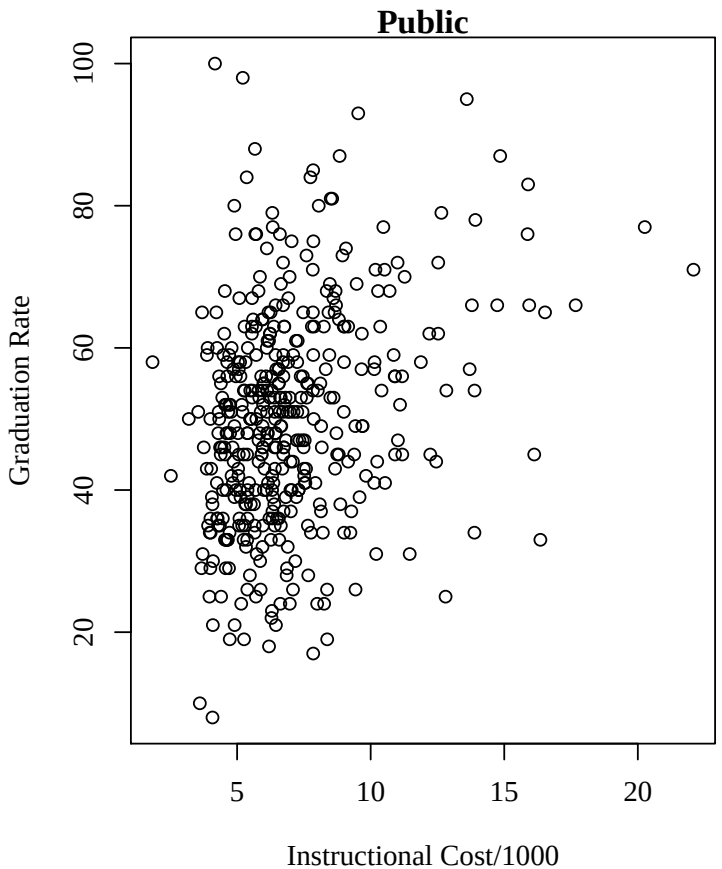
Supposing the model's theory is correct, we believe

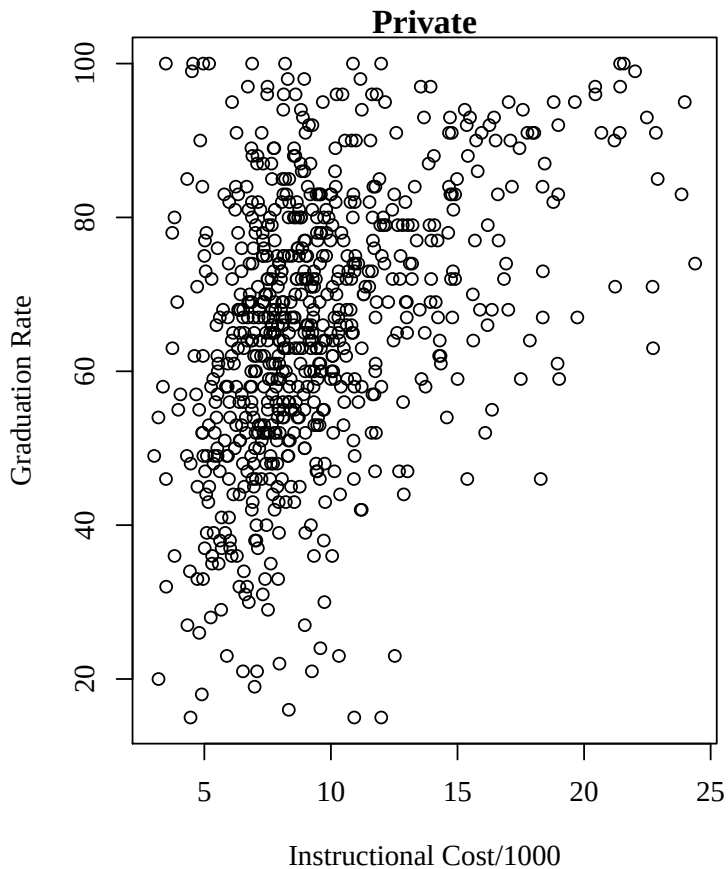
- We believe the true value of β_1 between "lwr" and "upr" with probability 0.95

Should we Subset the Data?

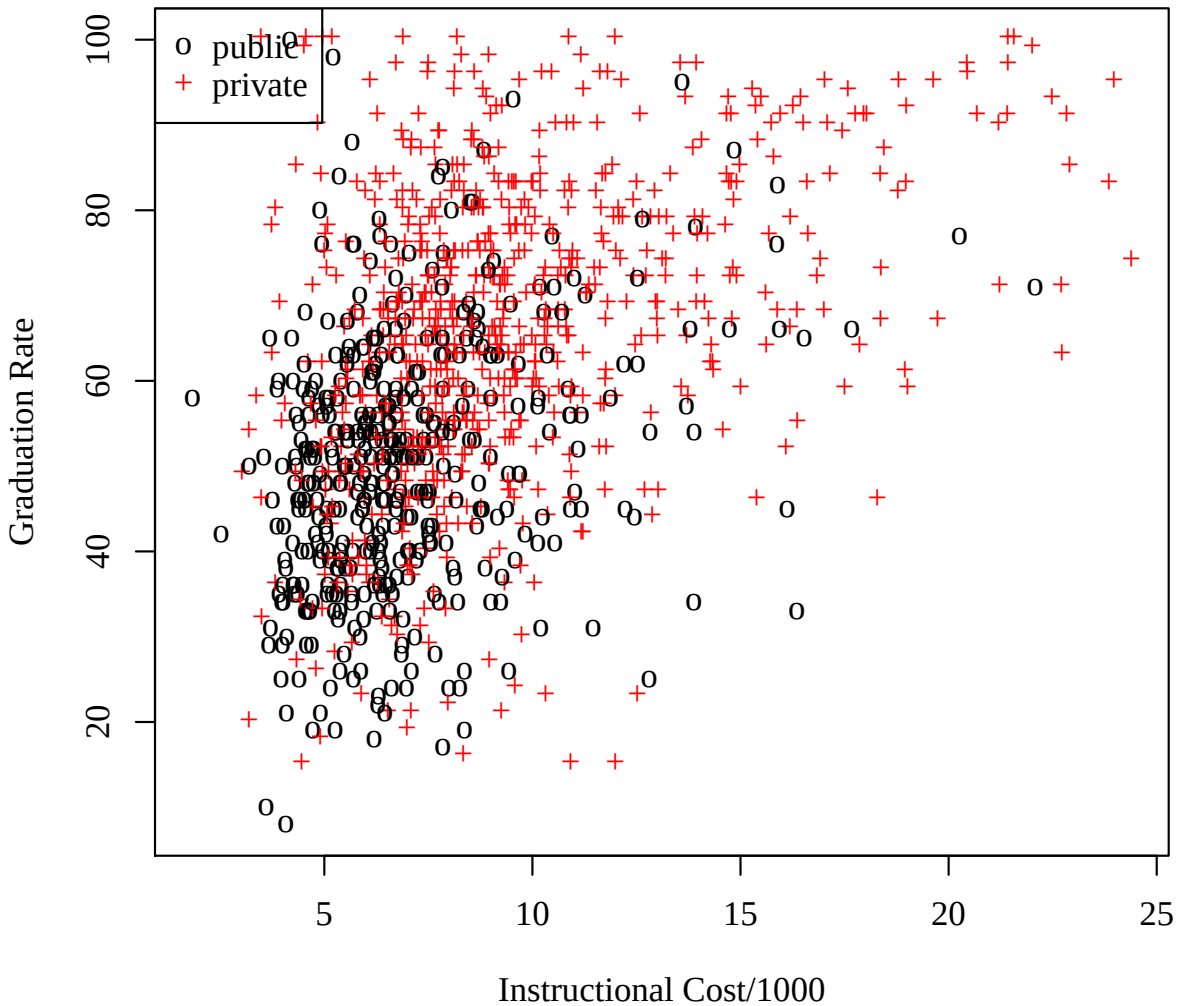
- Previous estimate pooled Public and Private Institutions
- If the relationships are different for the two types of schools, we have biased estimates of the effects

Confidence Intervals for Intercept and Slope

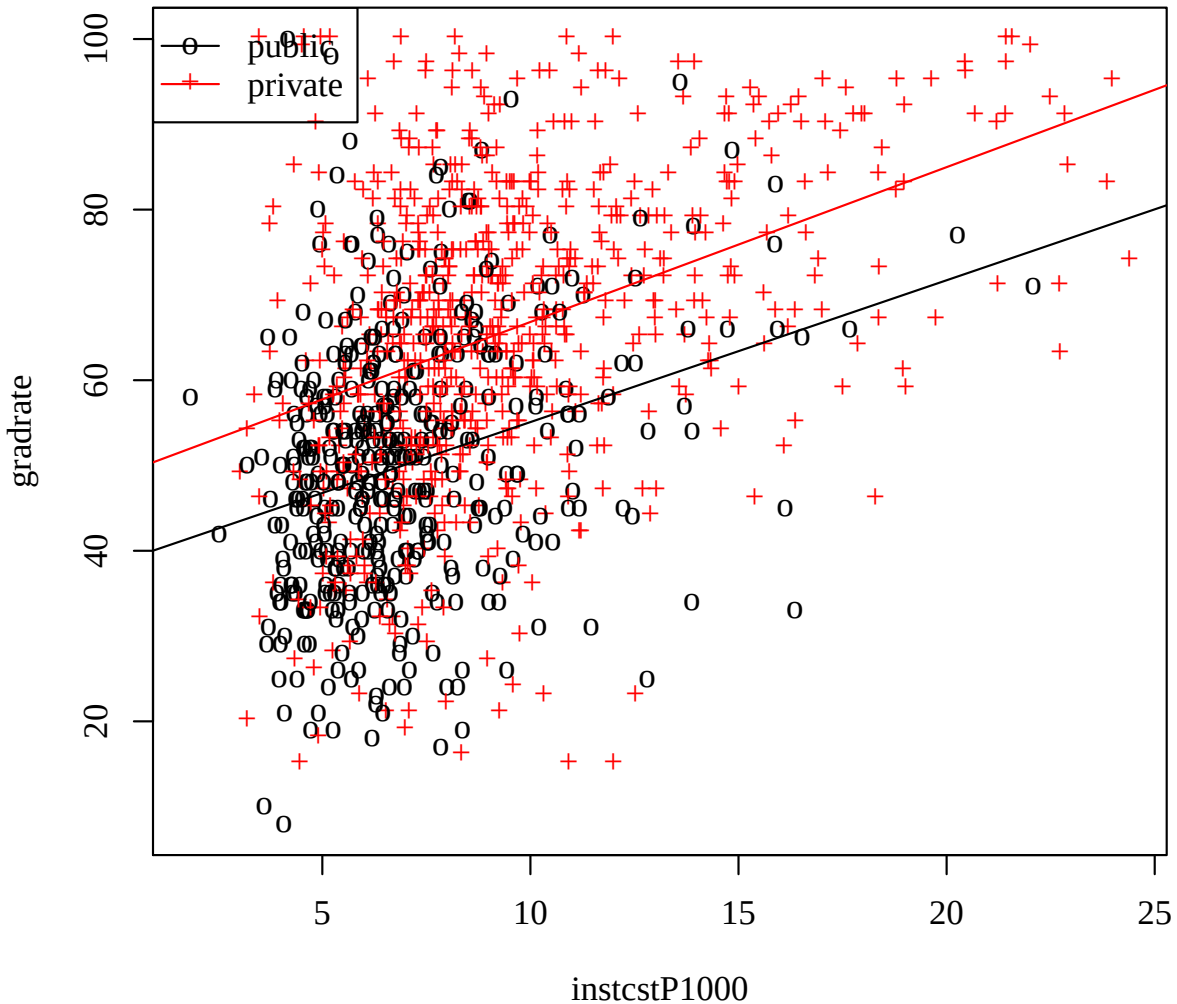




Superimpose and Color Code the 2 Types



Fit separate regressions, and superimpose the predicted values



Look at the Fits, Side by Side

	Public Estimate (S.E.)	Private Estimate (S.E.)
(Intercept)	38.459***	48.674***
	(1.978)	(1.607)
instcstP1000	1.663***	1.815***
	(0.263)	(0.157)
N	422	733
RMSE	14.598	16.124
R^2	0.087	0.154

$*p \leq 0.05$** $p \leq 0.01$***$p \leq 0.001$

Look forward to Multiple Regression

- This is not satisfactory.
  - We have no rigorous way to say the intercept and slope for the 2 school types are "statistically significantly different" from each other.
- But don't you wish you could say "the slopes are the same", but "the intercepts are different"?
- Those comparisons will have to wait, because they require the "multiple" in "multiple regression".