Lab 5: Regression

P. Johnson, B. Rogers, L. Shaw

Wednesday, February 25, 2015

Elementary (simple) regression part 2

- Pulling data from the internet
- Example
- Interpretation
- Mean centering
- Pleasing plots

Let's take a look at the publicspending.txt data.

- EX: Per capita state and local public expenditures ($)
- ECAB: Economic ability index, in which income, retail sales, and the value of output (manufactures, mineral, and agricultural) per capita are equally weighted.
- MET: Percentage of population living in standard metropolitan areas
- GROW: Percent change in population, 1950-1960
- YOUNG: Percent of population aged 5-19 years
- OLD: Percent of population over 65 years of age
- WEST: Western state (1) or not (0)
- STATE: Abbreviation of state

Get Data, put in same directory as this file

```
getwd()
```

```
## [1] "D:/Users/l076s857/Documents/GitHub/labstat/lab05"
```

```
if (!file.exists("publicspending.txt")){
    lab5.url <- "http://pj.freefaculty.org/guides/stat/DataSets/PublicSpending/publicspending.txt"
    download.file(lab5.url, destfile = "publicspending.txt")
}
lab5 <- read.table("publicspending.txt", header = TRUE)
```

- We first check to see if 'publicspending.txt' is in our working directory
- If publicspending is not found (!fileexists), run the two line between { and }
  - *assign internet url to a variable*
  - *download the file to working directory and call it publicspending.txt*
- Read data file from working directory

```
library(rockchalk)
```

```
## Warning: package 'rockchalk' was built under R version 3.0.3
```

Look at our data

- First, we want to check the names of all of our variables

```
head(lab5)
```

```
##    EX  ECAB  MET GROW YOUNG  OLD WEST STATE
## 1 256  85.5 19.7  6.9  29.6 11.0    0    ME
## 2 275  94.3 17.7 14.7  26.4 11.2    0    NH
## 3 327  87.0  0.0  3.7  28.5 11.2    0    VT
## 4 297 107.5 85.2 10.2  25.1 11.1    0    MA
## 5 256  94.9 86.2  1.0  25.3 10.4    0    RI
## 6 312 121.6 77.6 25.4  25.2  9.6    0    CT
```

Oops, that capitalization problem

- Last week, the data set on Blackboard had variables in UPPER CASE.
- You can convert them all to lower case with a single command for simpler typing.

```
colnames(lab5) <- tolower(colnames(lab5))
names(lab5)
```

```
## [1] "ex"    "ecab"  "met"   "grow"  "young" "old"   "west"  "state"
```

Look over the data

```
summarize(lab5)
```

```
## $numerics
##          ecab        ex     grow      met     old    west   young
## 0%    57.4000  183.0000  -7.4000   0.0000  5.4000  0.0000 24.0000
## 25%   85.4000  253.5000   6.9750  24.1000  7.9500  0.0000 26.4000
## 50%   95.3000  285.5000  14.0500  46.1500  9.4500  0.5000 28.0000
## 75%  105.1000  324.0000  22.6750  69.9750 10.4250  1.0000 29.6250
## 100% 205.0000  454.0000  77.8000  86.5000 11.9000  1.0000 32.9000
## mean  96.7542  286.6458  18.7292  46.1688  9.2125  0.5000 28.1146
## sd    22.2528   58.7948  18.8747  26.9388  1.6394  0.5053  2.1485
## var  495.1885 3456.8293 356.2562 725.6988  2.6875  0.2553  4.6162
## NA's   0.0000    0.0000   0.0000   0.0000  0.0000  0.0000  0.0000
## N     48.0000   48.0000  48.0000  48.0000 48.0000 48.0000 48.0000
##
## $factors
##             state
##  AL            : 1.000
##  AR            : 1.000
##  AZ            : 1.000
##  CA            : 1.000
##  (All Others) :44.000
##  NA's          : 0.000
##  entropy       : 5.585
##  normedEntropy: 1.000
##  N             :48.000
```

Request covariance matrix of whole data set

- We can also get a covariance matrix and correlation matrix of all of the variables in our dataset.

```
cov(lab5)
```

```
> cov(lab5)
Error: is.numeric(x) || is.logical(x) is not TRUE
```

```
cov(lab5[ , 1:6])
```

```
##                ex       ecab        met       grow      young        old
## ex    3456.829344 858.098316  71.646144 449.761613 -37.037278  -2.255053
## ecab   858.098316 495.188493 245.136835 193.237961 -28.182934  -1.623245
## met     71.646144 245.136835 725.698790 205.430931 -36.248258  -1.813005
## grow   449.761613 193.237961 205.430931 356.256152  -8.292562 -12.766330
## young  -37.037278 -28.182934 -36.248258  -8.292562   4.616166  -1.848910
## old     -2.255053  -1.623245  -1.813005 -12.766330  -1.848910   2.687500
```

And the correlation matrix

- We are not including our dichotomous variable, west

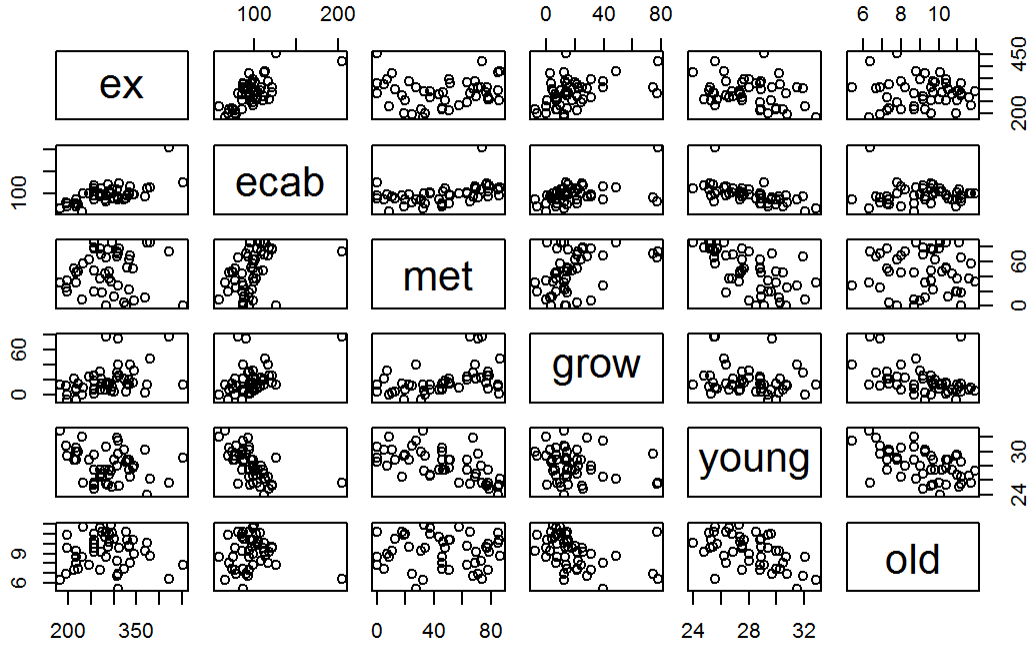
```
cor(lab5[ , 1:6])
```

```
##                ex        ecab         met       grow      young
## ex      1.00000000  0.65586251  0.04523511  0.4052866 -0.2931969
## ecab    0.65586251  1.00000000  0.40892636  0.4600722 -0.5894680
## met     0.04523511  0.40892636  1.00000000  0.4040233 -0.6262796
## grow    0.40528659  0.46007220  0.40402333  1.0000000 -0.2044875
## young -0.29319692 -0.58946801 -0.62627957 -0.2044875  1.0000000
## old   -0.02339611 -0.04449636 -0.04105316 -0.4125823 -0.5249292
##                old
## ex     -0.02339611
## ecab   -0.04449636
## met    -0.04105316
## grow   -0.41258234
## young  -0.52492921
## old     1.00000000
```

Plot matrix of bivariate relationships

- We are not plotting our dichotomous variable, west

```
plot(lab5[ , 1:6])
```



Elementary regression example

- public expenditures regressed on economic capabilities
- Step 1. Run the regression.

```
mod1 <-lm(ex ~ ecab, data = lab5)
```

- Step 2. Look at model summary.

```
summary(mod1)
```

```
##
## Call:
## lm(formula = ex ~ ecab, data = lab5)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66.29 -39.78  -7.83  35.29 117.02
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 118.9832    29.1802   4.078 0.000179 ***
## ecab          1.7329     0.2941   5.893 4.19e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.86 on 46 degrees of freedom
## Multiple R-squared:  0.4302, Adjusted R-squared:  0.4178
## F-statistic: 34.72 on 1 and 46 DF,  p-value: 4.193e-07
```

Look at vcov

```
vcov(mod1)
```

```
##             (Intercept)        ecab
## (Intercept)   851.48358 -8.36711041
## ecab           -8.36711  0.08647804
```

- You can use vcov info to compute model standard errors.

```
sqrt(diag(vcov(mod1)))
```

```
## (Intercept)        ecab
##  29.1801916   0.2940715
```

Model interpretation

- The average public expenditure (ex) is 118.98 when there is 0 economic capabilities (ecab).
- Does this interpretation make sense?
- What is public expenditure for average economic capabilities?

```
mean(lab5$ecab)
```

```
## [1] 96.75417
```

```
# b0 + b1ecab_at_mean
118.9832 + 1.7329*96.75417
```

```
## [1] 286.6485
```

Put centered predictor into regression

- We are centering at the mean, but we can choose any other value that would aid interpretation.
  - *If we were studying students in grades 1-5, an intercept when grade is 0 makes little sense.*
  - *We could recenter on the average grade, 3, so the intercept would be the average for 3rd graders (grade - mean(grade)).*
  - *We could 'recenter' grade so that intercept would be the average in grade 1 (grade - 1).*
- We could also put the variable on standard, normal scale with mean = 0 and variance = 1.
  - *This is based on the z-scale.*
  - *The regression coefficient is a standardized coefficient.*

Use scale to 'recenter'

- First list the variable you want to change.
- If you enter nothing else in the function, the variable will be standardized with mean = 0, variance = 1.
- Setting 'scale = FALSE' will not divide the term by its variance.
- If you want to center by something other than the mean, change 'center =' to your number.

```
lab5$ecabmc <- scale(lab5$ecab,  center = TRUE, scale = FALSE)
```

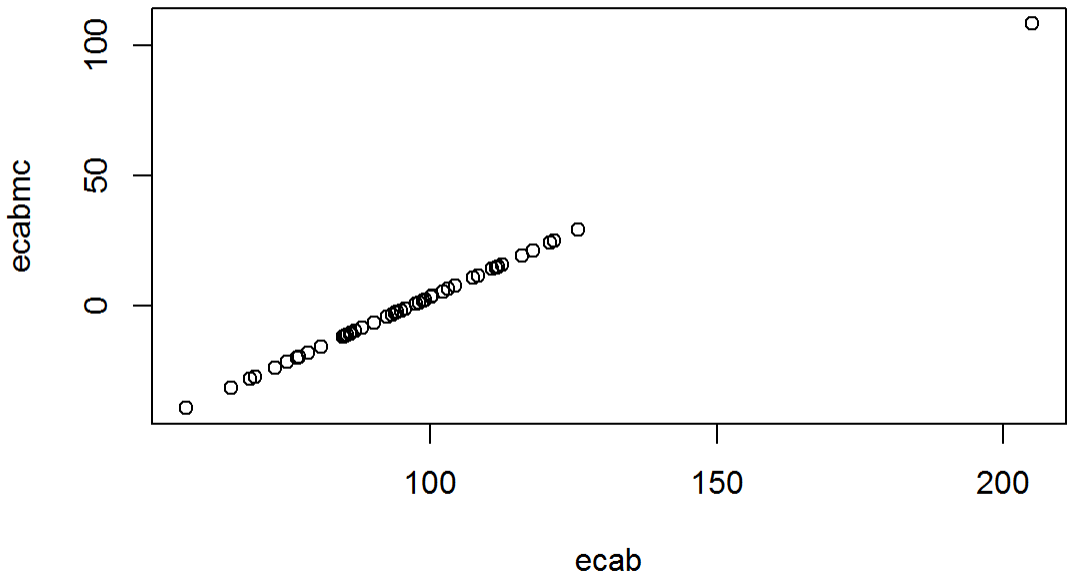
Check your work

- If rescaling worked, the original and rescaled variable will be perfectly correlated.

```
cor(lab5$ecab, lab5$ecabmc)
```

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

```
plot(ecabmc ~ ecab, lab5) # check your work
```



Re-run the model with ecabmc

```
mod2 <-lm(ex ~ ecabmc, data = lab5)
anova(mod2)
```

```
## Analysis of Variance Table
##
## Response: ex
##           Df Sum Sq Mean Sq F value    Pr(>F)
## ecabmc     1  69888   69888  34.724 4.193e-07 ***
## Residuals 46  92583    2013
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(mod2, mod1) # compare new with original
```

```
## Analysis of Variance Table
##
## Model 1: ex ~ ecabmc
## Model 2: ex ~ ecab
##   Res.Df   RSS Df Sum of Sq F Pr(>F)
## 1     46 92583
## 2     46 92583  0         0
```

Look at the mean centered model summary

- Recall from our manual calculation of our intercept at the mean of economic capability was 286.6485.

```
summary(mod2)
```

```
##
## Call:
## lm(formula = ex ~ ecabmc, data = lab5)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66.29 -39.78  -7.83  35.29 117.02
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 286.6458     6.4754  44.267  < 2e-16 ***
## ecabmc        1.7329     0.2941   5.893 4.19e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.86 on 46 degrees of freedom
## Multiple R-squared:  0.4302, Adjusted R-squared:  0.4178
## F-statistic: 34.72 on 1 and 46 DF,  p-value: 4.193e-07
```

Fast mean centering - rockchalk

```
summary(meanCenter(mod1))
```

```
## These variables were mean-centered before any transformations were made on the design matrix.
## NULL
## The centers and scale factors were
##
## mean
## scale
## The summary statistics of the variables in the design matrix (after centering).
##           mean std.dev.
## ex   286.64583 58.79481
## ecab  96.75417 22.25283
##
## The following results were produced from:
## meanCenter.default(model = mod1)
##
## Call:
## lm(formula = ex ~ ecab, data = stddat)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -66.29 -39.78  -7.83  35.29 117.02
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 118.9832    29.1802   4.078 0.000179 ***
## ecab          1.7329     0.2941   5.893 4.19e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.86 on 46 degrees of freedom
## Multiple R-squared:  0.4302, Adjusted R-squared:  0.4178
## F-statistic: 34.72 on 1 and 46 DF,  p-value: 4.193e-07
```

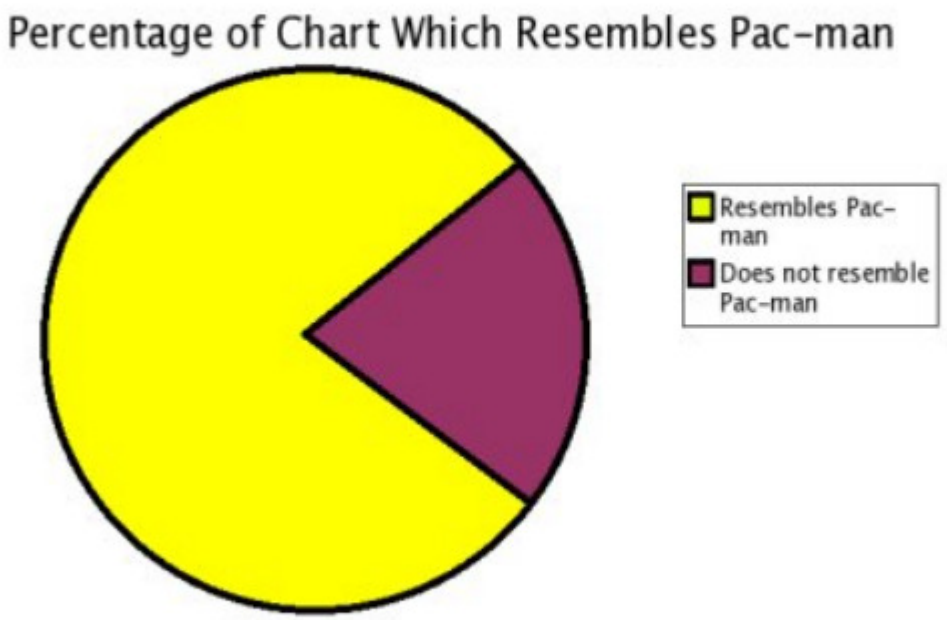
Pretty plots

- While default values for plot size are fine when getting a feel for your data, but what are some good practices when you need to create a publication ready figure?
- What type of plot should you use?
- What labels should you choose?
- How should you size the figure?
- Different R options
- Fonts and colors
- Saving to PDF

What type of plot should you use?

Pie charts

This is the only legitimate use for pie charts



Thomas Lumley,"Complex sampling and R" Presentation 2011-07-29 58

http://faculty.washington.edu/tlumley/survey-jsm-nup.pdf

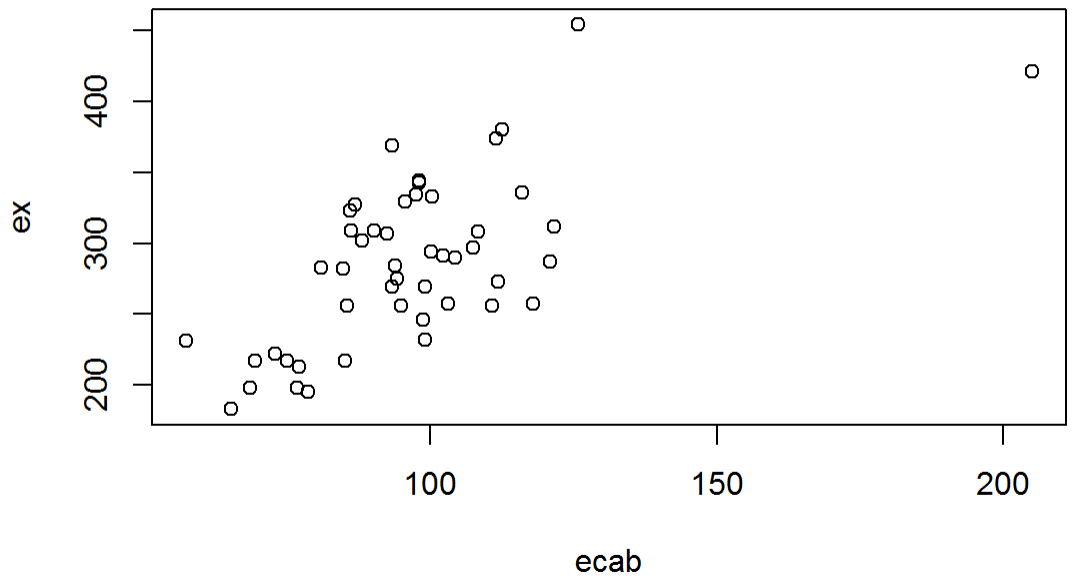
What type of plot should you use?

- Pie charts are never a good choice.
- Bar plots for categorical data and histograms are better ways to present frequency though tables are probably a better choice for descriptive statistics.
- Scatterplots are useful to illustrate how two variables are changing (or not changing) together or to highlight a non-linear relationship that has been previously hypothesized as linear.
- In a journal article, you will probably be limited on the number of figures you can submit with the article. Do you want to waste one on descriptive information?
- What type of figures do other people use in articles similar to the one you are writing?
- How is the picture going to help the reader understand the story you are telling?

What labels should you choose?

- By default, R provides no title and the variable names as the labels or other predefined values.

```
plot(ex ~ ecab, data = lab5)
```



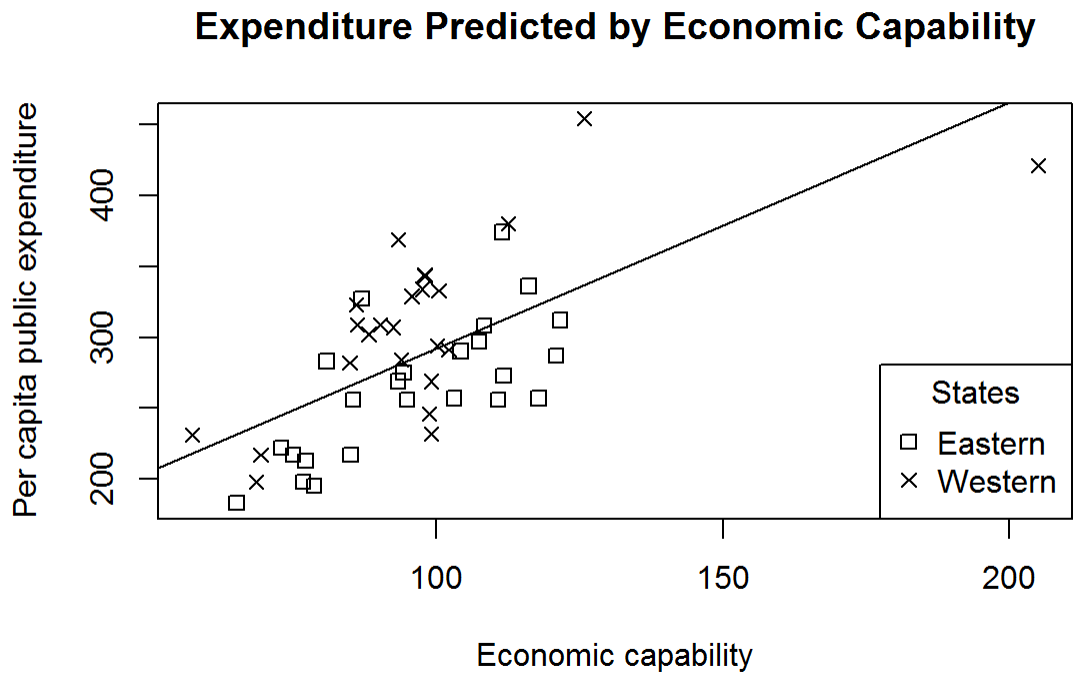
Labels

- Use xlab and ylab to change your axis labels to something that does not require knowledge of your data set to understand.
- Should you include a title within the plot itself? It depends on the publishing guidelines you are following. APA guidelines instructs you to crop any titles out so that the title is in the text of the document.
- If the plot is for a poster or other presentation, you might want to include a title.

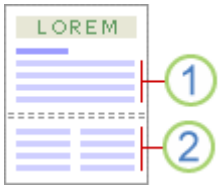
And a legend

- Legends can be placed inside a plot or in the margins with inset

```
plot(ex ~ ecab, xlab = "Economic capability", ylab = "Per capita public expenditure",
     main = "Expenditure Predicted by Economic Capability", type = "n", data = lab5)
with(lab5[lab5$west == 0, ], points(ecab, ex, pch = 0))
with(lab5[lab5$west != 0, ], points(ecab, ex, pch = 4))
abline(mod1)
legend("bottomright",legend=c("Eastern", "Western"), pch=c(0,4), title = "States")
```



How should you size the figure?

- You may have heard of aspect ratio, golden rectangle or the golden ratio.
  - *According to Edward Tufte "[g]raphics should tend toward the horizontal" (Tufte, E. R., 2001; The visual display of quantitative information, p. 186).*
  - *Tufte proposes two ratios: the golden rectangle–1/1.618 (the same used in Grecian temples)–or a simpler 1/1.5.*
  - *But you might want the two axes to be a little more proportional because it makes more sense in the context of what you are trying to present.*
  - *All plots in this presentation are 4 inches tall and 6 inches wide.*
- APA guidelines say to create a graphic no wider than 3.25 inches if the document will be in two columns or 6.875 inches wide if the document will be in 1 column. You may also be constrained to no wider that 6.5 inches for an 8.5 x 11 inch paper if you take into account 1 inch margins.
- Here is an example or one and two columns: 

Fonts and colors

- The default fonts and colors might not be what you want.
- If you are producing a figure to submit to a journal, check their guidelines on figures.
  - *Some journals will not accept figures with colors (or will send the bill to you for printing) so you will need to think about different shades of gray instead of colors.*
- Fonts need to be picked carefully as well.
  - *APA guidelines call for san serif fonts (Times New Roman has serifs and Arial is san serif); R defaults to a san serif font in its plots.*
  - *The font can range from 8-12 points. Make sure it is big enough to be readable.*

Different R options

- RStudio will place the figure in the Plots window, sized to what is on the screen.
- When you want to produce a higher quality graphic, set it up, view it, and then output the graphic directly to another format with size explicitly set.
- For better viewing when creating a graphic, use Emacs + R, Notepad++, or R itself.
    - *Creating a graphic will launch a separate graphics viewer.*

Saving to PDF

- To save a plot to an external file:

```
pdf("SampleGraph.pdf", width=7.5, height=5)
plot(ex ~ ecab, xlab = "Economic capability", ylab = "Per capita public expenditure",
     main = "Expenditure Predicted by Economic Capability", type = "n", data = lab5)
with(lab5[lab5$west == 0, ], points(ecab, ex, pch = 0))
with(lab5[lab5$west != 0, ], points(ecab, ex, pch = 4))
abline(mod1)
legend("bottomright",legend=c("Eastern", "Western"), pch=c(0,4), title = "States")
dev.off()
```