Outline

linear regression goal

Goal: Learn about an unknown *linear* function f that relates variables $Y \in \mathbb{R}$ and $X \in \mathbb{R}^p$ through the relationship:

$$Y = f(X) + \epsilon$$

The variables Y and X have distinct roles:

- Y : dependent variable
- X : independent variables (predictors, covariates, features)

four assumptions of linear regression

- linearity: linear relationship between predictors and response
- statistical independence of error
  - the error is not related to the variables
  - no correlation between consecutive errors (for time-series data)
- homoscedasticity: errors have constant variance
  - across time (in time series data)
  - across the predictions
- normality: error term is normally distributed

goals and terminology

Why do we do this:

Prediction: Predict the value of Y for a given point X (not necessarily a point X_i in the data).

Model inference: To learn about the relationship between X and Y, i.e. understanding which predictors or combinations of predictors are associated with particular changes in Y.

prediction vs. inference

- A model for the weather 48 hours from now based on current and historical weather...
  - could be very practical, accurate
  - would *not* provide insight into the atmospheric processes that underly changes in the weather.
- Model of relationship between childhood lead exposure and behavioral problems...
  - useful to assess if there is any risk due to lead exposure and estimate overall effects of lead exposure in a large population
  - effect of lead exposure on an individual child is probably too small in relation to numerous other risk factors to be of predictive value at the individual level

regression notation

For the training data (with p-predictors and n observations):

$$y_i \in \mathbb{R}$$

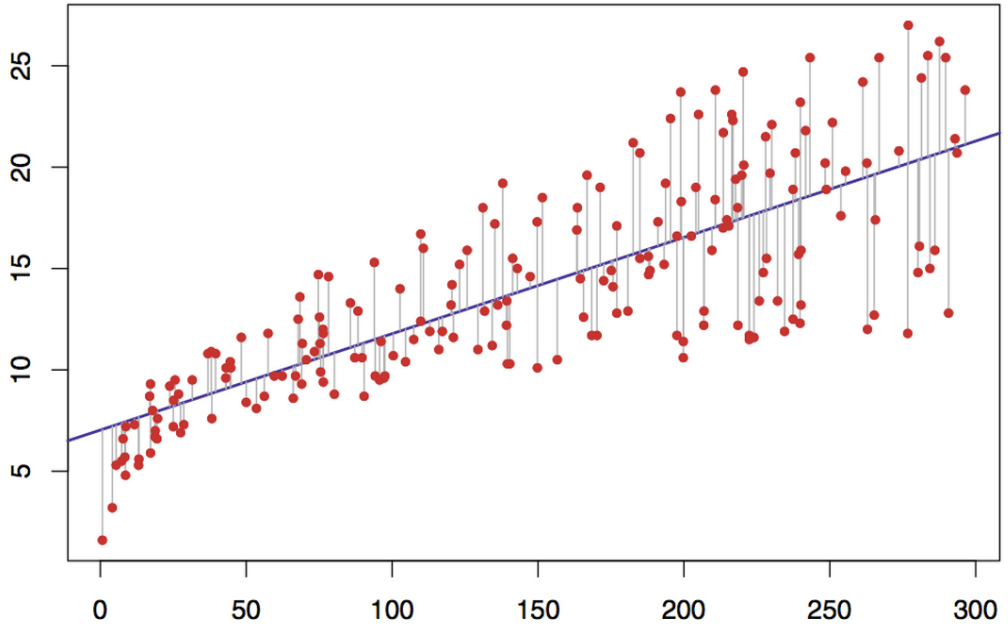
$$X_{\cdot i} = (X_{1i}, \ldots, X_{pi})' \in \mathbb{R}^p$$

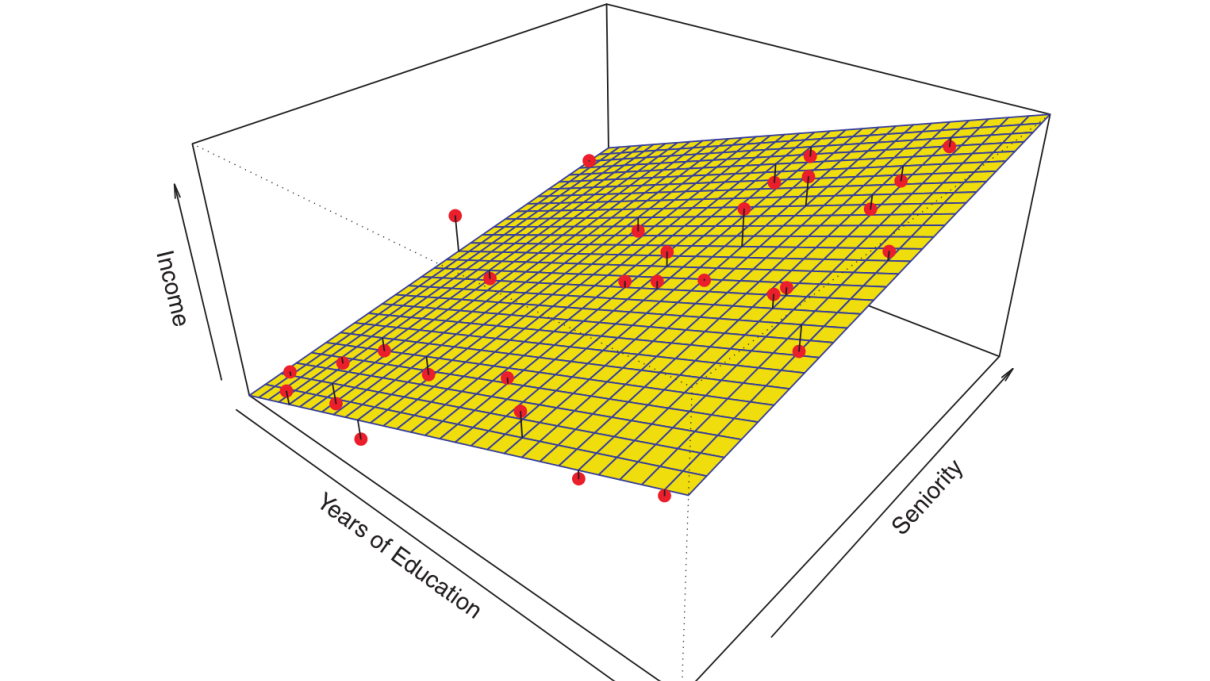
$$i = 1, \ldots, n$$

The linear function f takes the form:

$$y_i \approx f(X_{\cdot i}) = \beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \ldots + \beta_p X_{pi}$$

where $\beta_1, \ldots, \beta_p$ are the regression coefficients (partial slopes)


25
20
15
10
5
0
50
100
150
200
250
300


Income
Years of Education
Seniority

regression notation

A slight change to $X_{.i}$ (to clean up the notation): $X_{.i} = \begin{pmatrix} 1 \\ X_{1i} \\ \vdots \\ X_{pi} \end{pmatrix}$

and now we can write:

$$y_i \approx f(X_{.i}) = \beta' X_{.i}$$

where $\beta' = (\beta_0, \beta_1, \ldots, \beta_p)$ is a vector of regression coefficients

regression notation

For multiple regression ($p > 1$), the covariate data define the *design matrix*:

$$\mathbf{X} = \begin{pmatrix} 1 & X_{11} & X_{21} & \dots & X_{p1} \\ 1 & X_{12} & X_{22} & \dots & X_{p2} \\ & & \dots & & \\ & & \dots & & \\ & & \dots & & \\ 1 & X_{1n} & X_{2n} & \dots & X_{pn} \end{pmatrix}$$

predicted values

The vector of fitted values is given by the matrix-vector product

$$\hat{Y} = \mathbf{X}\beta$$

which is an *n*-dimensional vector.

The vector of residuals

$$Y - \mathbf{X}\beta = Y - \hat{Y}$$

is also an *n*-dimensional vector.

predicted values

Of course, we don't know β – if we did, we wouldn't need to "fit" the model.

We estimate β based on the observed data.

We denote the estimates for β as $\hat{\beta}$.

Therefore, our estimates are for Y are given by

$$\hat{Y} = \mathbf{X}\hat{\beta}$$

finding β

The goal of least-squares estimation is to minimize the sum of squared differences between the fitted and observed values.

$$L(\hat{\beta}) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = ||Y - \mathbf{X}\hat{\beta}||_2^2$$

Estimating β by minimizing $L(\hat{\beta})$ is called method of ordinary least squares (OLS).

finding β

Finding $\hat{\beta}$ that minimizes the sum of squared error has a closed-form, analytical solution:

(i.e. differentiate $L(\hat{\beta})$ w.r.t. $\hat{\beta}$, set to 0, solve the $p+1$ system of equations)

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$$

The matrix $\mathbf{X}'\mathbf{X}$ is not always invertible. Two common conditions that cause this are:

- two or more of the predictors are highly correlated (multicollinearity)
- there are more predictors than observations ($p > n$)

variance of β

The estimate for the variance of $\hat{\beta}$, $\hat{\text{Var}}(\hat{\beta})$ is computed from

$$\hat{\sigma}^2 \mathbf{X}'\mathbf{X}$$

where $\hat{\sigma}^2$ is unbiased estimate of the residual variance

$$\frac{1}{n-p-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

demonstration with R

Franco Modigliani hypothesizes that the saving ratio (aggregate personal saving divided by disposable income) for nations can be explained based on 4 variables.

`LifeCycleSavings` data frame contains 50 observations.

sr	savings ratio
pop15	% of population under 15
pop75	% of population over 75
dpi	real per-capita disposable income
ddpi	% growth rate of dpi

demonstration with R

```
data(LifeCycleSavings)
head(LifeCycleSavings)
```

	sr	pop15	pop75	dpi	ddpi
Australia	11.43	29.35	2.87	2329.68	2.87
Austria	12.07	23.32	4.41	1507.99	3.93
Belgium	13.17	23.8	4.43	2108.47	3.82
Bolivia	5.75	41.89	1.67	189.13	0.22
Brazil	12.88	42.19	0.83	728.47	4.56
Canada	8.79	31.72	2.85	2982.88	2.43

demonstration with R

```
lm(data=LifeCycleSavings,
      sr ~ pop15 + pop75 + dpi + ddpi)
```

finding β

```
Y<-LifeCycleSavings$sr

X<-as.matrix(LifeCycleSavings[,2:5])
X<-cbind(intercept=1,X)

head(X)
```

	intercept	pop15	pop75	dpi	ddpi
Australia	1	29.35	2.87	2329.68	2.87
Austria	1	23.32	4.41	1507.99	3.93
Belgium	1	23.8	4.43	2108.47	3.82
Bolivia	1	41.89	1.67	189.13	0.22
Brazil	1	42.19	0.83	728.47	4.56
Canada	1	31.72	2.85	2982.88	2.43

finding β

Operation	Description
A * B	element-wise multiplication
A %*% B	matrix multiplication
A %o% B	Outer product. AB'
t(A)	transpose
diag(A)	returns principal diagonal elements
solve(A)	inverse of A

finding β

X′X is coded as: `t(X)%*%X`

	intercept	pop15	pop75	dpi	ddpi
intercept	50	1754.48	114.65	55337.92	187.88
pop15	1754.48	65667.953	3497.171	1605780.91	6531.085
pop75	114.65	3497.171	344.5309	176211.33	435.405
dpi	55337.92	1605780.914	176211.3273	109354944.4	189895.306
ddpi	187.88	6531.085	435.405	189895.31	1109.55

finding β

$\mathbf{X}'\mathbf{X}^{-1}$ is coded as: `solve(t(X)%*%X)`

	intercept	pop15	pop75	dpi	ddpi
intercept	3.74E+00	-7.24E-02	-4.46E-01	-7.86E-05	-1.88E-02
pop15	-7.24E-02	1.45E-03	8.30E-03	1.68E-06	2.01E-04
pop75	-4.46E-01	8.30E-03	8.12E-02	-2.56E-05	-8.05E-04
dpi	-7.86E-05	1.68E-06	-2.56E-05	6.00E-08	3.23E-06
ddpi	-1.88E-02	2.01E-04	-8.05E-04	3.23E-06	2.66E-03

finding β

$\beta = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$ is coded as:

```
beta<-solve(t(X)%*%X)%*%t(X))%*%Y
```

```
> beta
```

	[,1]
intercept	28.566087
pop15	-0.461193
pop75	-1.691498
dpi	-0.000337
ddpi	0.409695

finding β

`fit<-lm(data=LifeCycleSavings, sr ~ pop15 + pop75 + dpi + ddpi)`

Coefficients:

	Estimate
(Intercept)	28.5660865
pop15	-0.4611931
pop75	-1.6914977
dpi	-0.0003369
ddpi	0.4096949

finding the standard error of β

residual sum of squares: $\sum_{i=1}^{50}(y_i - \hat{y}_i)^2$

```
> sum(fit$residuals^2)
[1] 650.713
```

residual variance: $\frac{\text{rss}}{50-4-1}$ $\hat{\sigma}^2 = \frac{650.713}{45} = 14.46029$

finding the standard error of β

the standard error of $\hat{\beta}$ is just diagonal elements of the square root of $\hat{\text{Var}}(\hat{\beta})$

$$\text{diag}\left(\hat{\sigma}\sqrt{\mathbf{X}'\mathbf{X}}\right)$$

```
> diag(3.803 *sqrt(solve(t(X)%*%X)))
  intercept       pop15       pop75         dpi        ddpi
7.3551569540 0.1446548284 1.0836933519 0.0009311883 0.1962142236
```

demonstration with `lm` in R

`fit<-lm(data=LifeCycleSavings, sr ~ pop15 + pop75 + dpi + ddpi)`

Coefficients:

	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)	
(Intercept)	28.5660865	7.3545161	3.884	0.000334	***
pop15	-0.4611931	0.1446422	-3.189	0.002603	**
pop75	-1.6914977	1.0835989	-1.561	0.125530	
dpi	-0.0003369	0.0009311	-0.362	0.719173	
ddpi	0.4096949	0.1961971	2.088	0.042471	*

```
Call:
lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2422 -2.6857 -0.2488  2.4280  9.7509

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
pop15       -0.4611931  0.1446422  -3.189 0.002603 **
pop75       -1.6914977  1.0835989  -1.561 0.125530
dpi         -0.0003369  0.0009311  -0.362 0.719173
ddpi         0.4096949  0.1961971   2.088 0.042471 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.803 on 45 degrees of freedom
Multiple R-squared:  0.3385, Adjusted R-squared:  0.2797
F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904
```

interpretation of β

The regression coefficient for the intercept, β_0, is the value you would predict if all other predictors were equal to 0.

Regression coefficients for the predictors represent the change in the outcome variable for a *one unit of change* of that predictor variable – *holding all other predictors constant*.

e.g., holding other variables constant, for each unit increase of X_1, we expect Y to increase by β_1

interpretation example

Coefficients:

	Estimate
(Intercept)	28.5660865
pop15	-0.4611931
pop75	-1.6914977
dpi	-0.0003369
ddpi	0.4096949

- if entire population is 15 to 75, real per-capita disposable income and its growth rate is 0 $\rightarrow$ savings ratio is 28.56
- for every 1% point increase in population under 15, saving ratio drops by -0.46
- if over-75 population increases, serious hit to `sr`
- very small effect on `sr` for a one unit change in `dpi` – *what is the scale?*
- a positive relationship between savings ratio and the growth rate of `dpi`

feature selection

All variables that are included in an OLS model will have estimates for their regression coefficients – *even if they should not be in the model*.

Not all of the predictors had values for β that were significant.

If we remove predictors, then all of the coefficient estimates will be adjusted.

We can remove (or add) predictors *one at a time* and watch how the regression coefficients and assessment metrics(i.e., AIC, BIC, Adjusted R^2) change each time.

let's update the model...

	Estimate	Std. Error	t value	Pr($\geq |t|$)	
(Intercept)	28.5660865	7.3545161	3.884	0.000334	***
pop15	-0.4611931	0.1446422	-3.189	0.002603	**
pop75	-1.6914977	1.0835989	-1.561	0.125530	
dpi	-0.0003369	0.0009311	-0.362	0.719173	
ddpi	0.4096949	0.1961971	2.088	0.042471	*

`Multiple R-squared: 0.3385, Adjusted R-squared: 0.2797`

`AIC(fit): 282.1961`
`BIC(fit): 293.6683`

Since, `dpi` has the largest p-value, let's look at a revised model without it.

let's update the model...

```
fit2<-update(fit,~.-dpi)
```

	Estimate	Std. Error	t value	Pr($\geq |t|$)	
(Intercept)	28.125	7.184	3.92	0.0003	***
pop15	-0.452	0.141	-3.21	0.0025	**
pop75	-1.835	0.998	-1.84	0.0725	.
ddpi	0.428	0.188	2.28	0.0275	*

```
Multiple R-squared: 0.337, Adjusted R-squared: 0.293
AIC(fit2): 280.3414
BIC(fit2): 289.9015
```

let's update the model...

```
fit3<-update(fit2,~.-pop75)
```

	Estimate	Std. Error	t value	Pr($\geq |t|$)	
(Intercept)	15.5996	2.3344	6.68	2.5e-08	***
pop15	-0.2164	0.0603	-3.59	0.0008	***
ddpi	0.4428	0.1924	2.30	0.0258	*

Multiple R-squared: 0.2878, Adjusted R-squared: 0.2575

AIC(fit3): 281.8861
BIC(fit3): 289.5342

let's update the model...

What if we added variables based on the interactions?

```
fit4<-lm(data=LifeCycleSavings,
         sr ~ pop15 * pop75 * dpi * ddpi)
```

	Estimate	Std. Error	t value	Pr($\geq |t|$)
(Intercept)	-0.0882202	44.6522904	-0.002	0.998
pop15	0.081951	1.0771182	0.076	0.94
pop75	8.5513275	17.4312432	0.491	0.627
dpi	-0.0061387	0.0415464	-0.148	0.883
ddpi	4.5959209	14.1652647	0.324	0.748
pop15:pop75	-0.1674024	0.4723558	-0.354	0.725
pop15:dpi	0.0003742	0.001081	0.346	0.731
pop75:dpi	0.0025893	0.0137749	0.188	0.852
pop15:ddpi	-0.052364	0.3289468	-0.159	0.874
pop75:ddpi	-1.7296781	4.9887381	-0.347	0.731
dpi:ddpi	0.0051729	0.0116381	0.444	0.66
pop15:pop75:dpi	-0.0001962	0.0003976	-0.493	0.625
pop15:pop75:ddpi	0.0145471	0.1203934	0.121	0.905
pop15:dpi:ddpi	-0.0002277	0.000309	-0.737	0.466
pop75:dpi:ddpi	-0.0022005	0.0041709	-0.528	0.601
pop15:pop75:dpi:ddpi	0.0001109	0.0001244	0.891	0.379

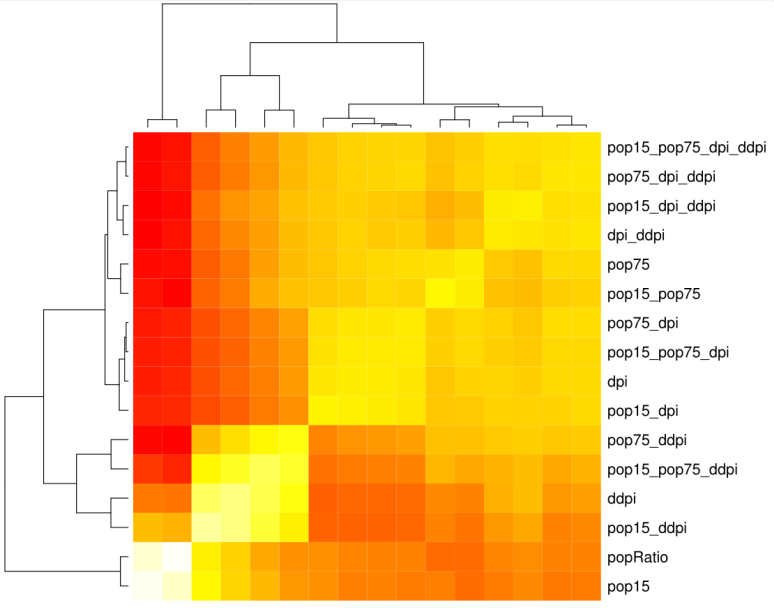
let's update the model...

```
Multiple R-squared:  0.463, Adjusted R-squared:  0.2261
F-statistic: 1.954 on 15 and 34 DF,  p-value: 0.05229

AIC(fit4): 293.7691
BIC(fit4): 326.2735
```

When too many variables are included in a model they can mask the truly significant ones!

heatmap for variable correlation



PCA on data

An excerpt from the PCA summary for the data including all interactions...

	PC1	PC2	PC3	PC4
Standard deviation	3.2007	1.9010	1.03487	0.71950
Proportion of Variance	0.6403	0.2259	0.06694	0.03235
Cumulative Proportion	0.6403	0.8662	0.93310	0.96545

multicollinearity

Multicollinearity refers to strong linear dependence between some of the covariates in a multiple regression model.

The usual interpretation of the regression coefficients is lost. The value and even sign of the coefficient can change drastically!

The large coefficient standard errors reflect the large uncertainty about the value of β.

Does not impact the predictive accuracy.

multicollinearity: example

Let X_1 be a normally distributed variable

Let $X_2 = 10X_1 + \epsilon$

(X_1 and X_2 are highly correlated!)

$Y = X_1 + \epsilon$

And $n = 100$

multicollinearity

`lm(formula = y ~ x1 + x2)`

	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)
(Intercept)	0.0028	0.0092	0.306	0.760
x1	47.4162	99.2767	0.478	0.634
x2	-4.6417	9.9277	-0.468	0.641

`Multiple R-squared: 0.9907, Adjusted R-squared: 0.9906`
`F-statistic: 5190 on 2 and 97 DF, p-value: < 2.2e-16`

The model recovered is:

$$Y \approx 47.4162 X_1 - 4.6417 X_2$$

multicollinearity

Multicollinearity is not necessarily fatal, but you need to know that it is there.

- It affects model interpretation more than model accuracy
- Introducing interaction terms will often cause multicollinearity

If you discover multicollinearity:

- Understand that the β are not true marginal effects
- Consider dropping variables to obtain a simpler model
- Expect to see big standard errors on your coefficients (i.e., your coefficient estimates are unstable)

results so far...

Model	R obj	Adj R^2	AIC	BIC
main effects full	`fit`	0.2797	282.1961	293.6683
main effects - `dpi`	`fit2`	0.2933	280.3414	289.9015
main effects - `dpi`, `pop75`	`fit3`	0.2575	281.8861	289.5342
main and all interactions	`fit4`	0.2261	293.7691	326.2735

stepwise variable selection

Variable selection: Use a subject-matter expertise and/or a lot of toying around with the possibilities...

However, there are also automatic methods that do this...

stepwise variable selection

backwards elimination	start: full model; eliminate vars one at a time
forward selection	start: empty model; add vars one at a time
stepwise regression	add and eliminate vars as needed

Common criteria: F-test, p-values, R^2, R^2_{adj}, AIC, BIC

Please note: stepwise selection is somewhat controversial

stepwise variable selection

The function `stepAIC` from `MASS` performs backwards, forwards, or stepwise using a modified AIC criteria

```
fit4<-lm(data=LifeCycleSavings,
         sr ~ pop15 * pop75 * dpi * ddpi)

fit5 <- stepAIC(fit4, direction="both")
fit5$anova #display results
```

```
> fit5$anova
Stepwise Model Path
Analysis of Deviance Table

Initial Model:
sr ~ pop15 * pop75 * dpi * ddpi

Final Model:
sr ~ pop15 + dpi + ddpi + dpi:ddpi


                     Step Df   Deviance Resid. Df Resid. Dev      AIC
1                                             34   528.2282 149.8753
2  - pop15:pop75:dpi:ddpi  1 12.3452049         35   540.5735 149.0304
3        - pop15:dpi:ddpi  1  1.5615558         36   542.1350 147.1746
4       - pop15:pop75:dpi  1 17.9359333         37   560.0709 146.8020
5             - pop15:dpi  1  0.1412616         38   560.2122 144.8146
6      - pop15:pop75:ddpi  1 13.7635576         39   573.9758 144.0282
7            - pop15:ddpi  1  2.6210042         40   576.5968 142.2560
8        - pop75:dpi:ddpi  1  2.2859686         41   578.8827 140.4538
9            - pop75:ddpi  1  6.5457716         42   585.4285 139.0161
10            - pop75:dpi  1 10.1679218         43   595.5964 137.8770
11         - pop15:pop75  1  2.9582574         44   598.5547 136.1247
12                - pop75  1 16.7083205         45   615.2630 135.5013
```

stepwise variable selection

We can check out this recommended model:

	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)	
(Intercept)	16.5287997	4.3729241	3.780	0.000459	***
pop15	-0.2023669	0.0981090	-2.063	0.044943	*
dpi	-0.0027411	0.0011774	-2.328	0.024457	*
ddpi	0.0462479	0.2439993	0.190	0.850521	
dpi:ddpi	0.0008171	0.0003593	2.274	0.027802	*

```
Multiple R-squared: 0.3745, Adjusted R-squared: 0.3189
```

```
AIC(fit5): 279.3952
BIC(fit5): 290.8673
```

stepwise variable selection

- stepwise selection is very common
- but it is in no way a “silver-bullet”
- it does not guarantee to select the “best” one among all of the possible models
- some would classify it as *data dredging*
- AIC stepwise selection (used in R) is probably better than p-value stepwise selection (the default in SAS)
- there are better automatic methods out there...

data dredging

data dredging, also called data-fishing, hypothesis-fishing, data-snooping, or *p*-hacking, is the practice of misusing data mining techniques to show misleading research

Image taken from Young, S. and A. Karr. “Deming, data, and observational studies”, *Signifigance*, September, 2011.

Deer in Headlights. A deer caught in the headlights will freeze, much like an author or reader seeing a p-value < 0.05, and think there must be a real effect. Authors can exploit this phenomenon intentionally or fool both themselves and the reader. Illustration: Tom Boulton

we can still update the model...

```
fit6<-update(fit5,~.-ddpi)
```

	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)	
(Intercept)	16.6320849	4.2931314	3.874	0.000337	***
pop15	-0.2008947	0.0967707	-2.076	0.043512	*
dpi	-0.0028701	0.0009505	-3.020	0.004120	**
dpi:ddpi	0.0008596	0.0002776	3.097	0.003329	**

```
Multiple R-squared: 0.374, Adjusted R-squared: 0.3332

AIC(fit6): 277.4351
BIC(fit6): 286.9952
```

we can still update the model...

Can we do better?

Let's create a new feature: $\texttt{popRatio} = \dfrac{\texttt{pop15}}{\texttt{pop75}}$

and then "search" for a better model... such as:

```
fit7<-lm(data=LifeCycleSavings,
         sr ~ pop15 + dpi + dpi:ddpi + popRatio)
```

Model	R obj	Adj R^2	AIC	BIC
main effects full	`fit`	0.2797	282.1961	293.6683
main effects - `dpi`	`fit2`	0.2933	280.3414	289.9015
main effects - `dpi`, `pop75`	`fit3`	0.2575	281.8861	289.5342
main and all interactions	`fit4`	0.2261	293.7691	326.2735
stepwise	`fit5`	0.3189	279.3952	290.8673
stepwise - `ddpi`	`fit6`	0.3332	277.4351	286.9952
new feature	`fit7`	0.3698	275.5102	286.9824

interactions and dummy coding

Another example...

Explore relationship between log wage rate (log(income/hours)) with predictors from the 2000 census data.
(data available in course website).

Predictors include:

- Age, Gender
- Education: 9 levels from none to PhD.
- Marital status: married, divorced, separated, or single.
- Race: white, black, Asian, other.

```
> C <- read.csv("~/Desktop/census2000.csv")
> head(C)
```

	age	sex	marital	race	education	income	hours
1	48	M	Married	White	3.hsgrad	52000	2600
2	24	M	Divorced	White	2.high	35000	2080
3	19	F	Single	Black	3.hsgrad	2400	240
4	18	M	Single	Black	2.high	6100	1500
5	28	M	Married	Other	4.assoc	22000	2080
6	18	F	Single	Black	3.hsgrad	2400	700

A good habit: build a dataframe with potentially relevant variables:

```
YX <- data.frame(log.WR = log(C$income/C$hours))
YX$age <- C$age
YX$age2 <- C$age^2
YX$sex <- C$sex
```

Next, use relevel to make “White” and “Married” the reference level

```
YX$race <- relevel(C$race, "White")
YX$marital <- relevel(C$marital, "Married")
```

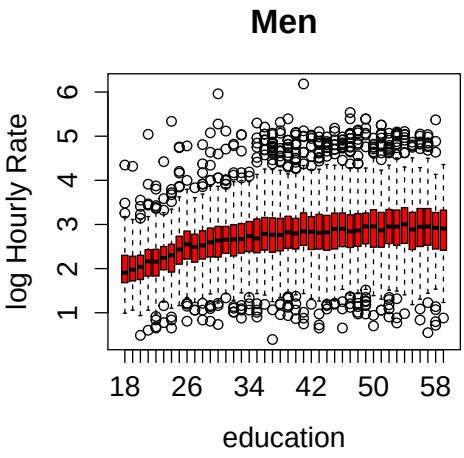
Now we create a several education indicator variables:

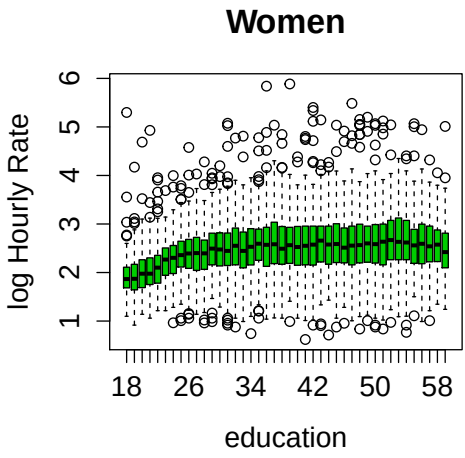
```
YX$hs <- C$edu == "3.hsgrad"
YX$assoc <- C$edu == "4.assoc"
YX$coll <- C$edu == "5.bachs"
YX$grad <- as.numeric(C$edu) > 6
```

and filter to only include “workers” ,i.e., hours $>$ 500, income $>$ \$5000, age $<$ 60

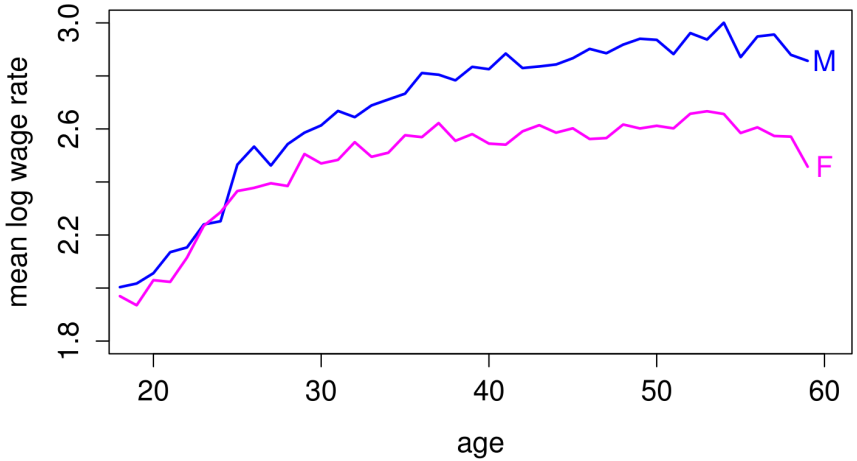
```
YX <- YX[(C$hours > 500) & (C$income > 5000) & (C$age < 60), ]
```

Explore the relationship between log wage rate and age – a proxy for experience. (parallel boxplots based on age)



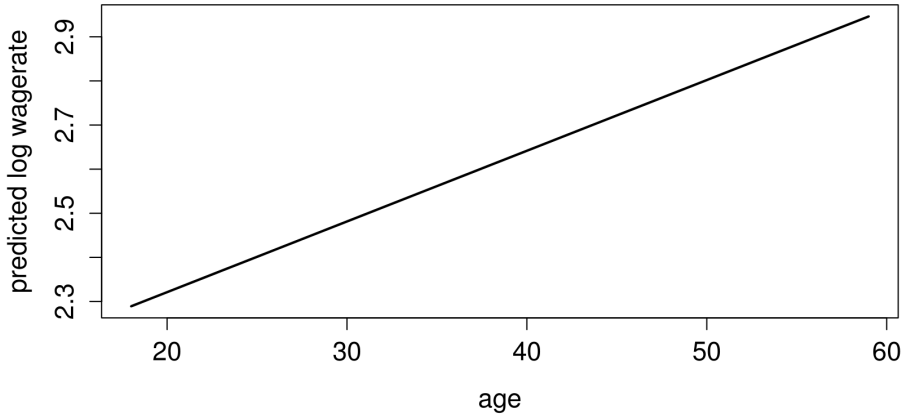


Notice the discrepancy between mean log(WR) for men and women: female wages flatten at about 30, while men's keep rising.



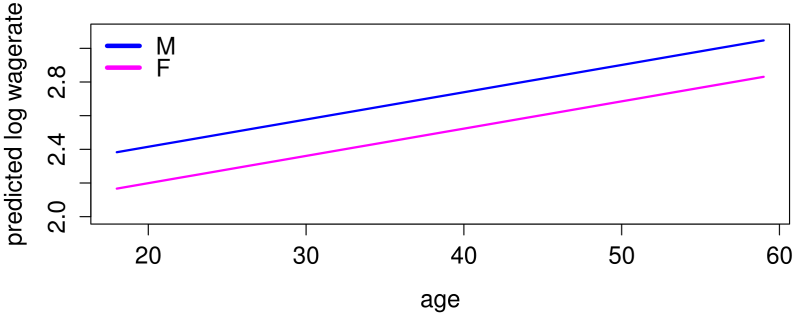
The simplest model:

`lm(log.WR ~ age, data = YX)`



You get one line for both men and women.

Add a gender effect: `lm(log.WR ~ age + sex, data=YX)`



	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)	
(Intercept)	1.8749712	0.0147775	126.88	$\leq 2^{-16}$	***
age	0.0162042	0.0003513	46.12	$\leq 2^{-16}$	***
sexM	0.2162319	0.0075217	28.75	$\leq 2^{-16}$	***

interpretation

	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)	
(Intercept)	1.8749712	0.0147775	126.88	$\leq 2^{-16}$	***
age	0.0162042	0.0003513	46.12	$\leq 2^{-16}$	***
sexM	0.2162319	0.0075217	28.75	$\leq 2^{-16}$	***

Essentially, the “dummy variable” allows us to have two models: one for men, another for women.

$$\textbf{for Women: } Y = 1.875 + 0.0162\, X_1$$
$$\textbf{for Men: } Y = 2.091 + 0.0162\, X_1$$

where Y is log wage rate, X_1 is age

interpretation

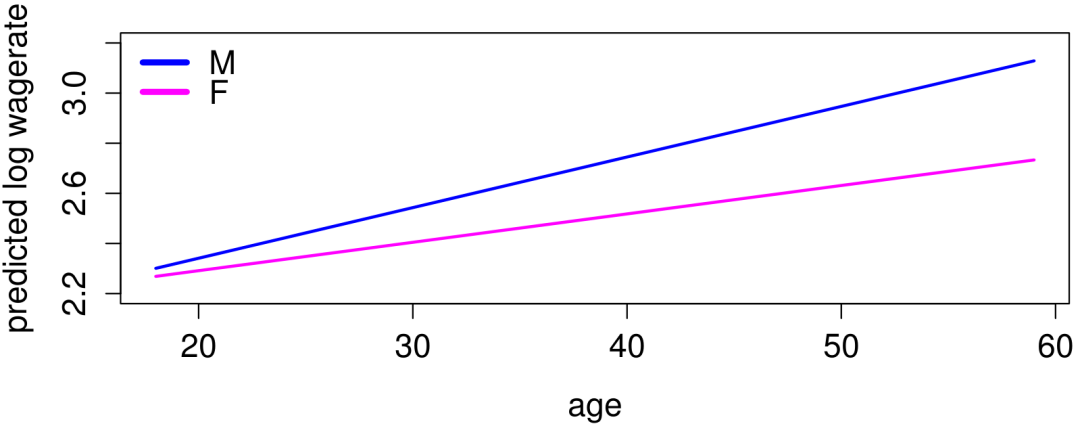
Add interactions: `lm(log.WR ~ age*sex, data=YX)`

	Estimate	Std. Error	t value	Pr($\geq \lvert t \rvert$)	
(Intercept)	2.0648606	0.0210849	97.931	$\leq 2^{-16}$	***
age	0.0113298	0.0005221	21.700	$\leq 2^{-16}$	***
sexM	-0.1275018	0.0283166	-4.503	6.74e-06	***
age:sexM	0.0088621	0.0007040	12.588	$\leq 2^{-16}$	***

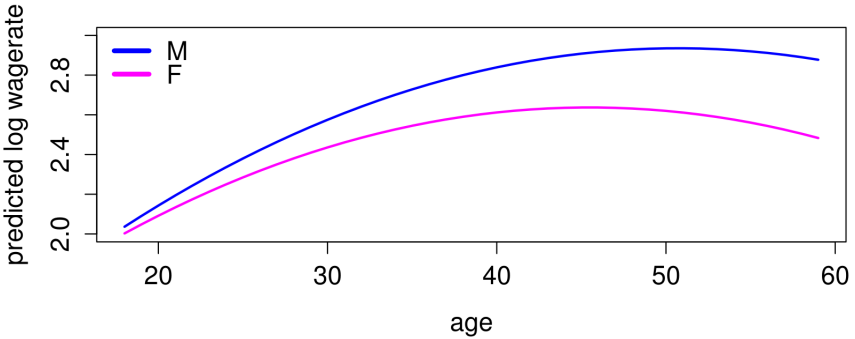
for Women: $Y = 2.065 + 0.0113\ X_1$

for Men: $Y = 1.937 + 0.0202\ X_1$

where Y is log wage rate, X_1 is age


predicted log wagerate
2.2
2.6
3.0
M
F
20
30
40
50
60
age

Add quadratic term: `lm(log.WR ~ age*sex + age2, data=YX)`

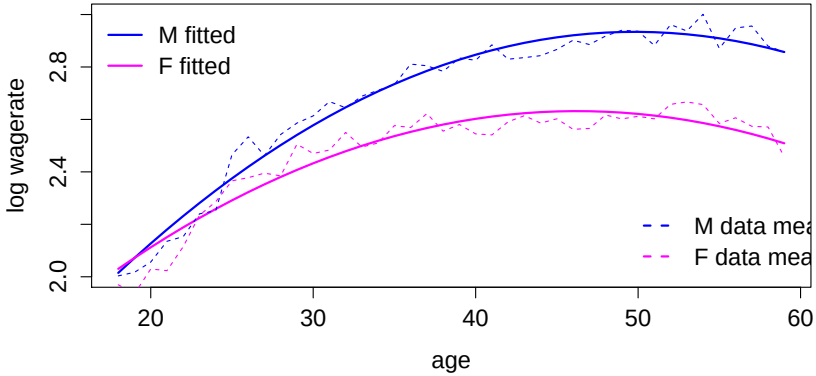


for Women: $Y = 0.8989 + 0.0765\, X_1 - 0.0008\, X_1^2$

for Men: $Y = 0.7743 + 0.0853\, X_1 - 0.0008\, X_1^2$

Add interaction with quadratic term:

`lm(log.WR ~ age*sex + age2*sex, data=YX)`



Modeling...

How many other models might fit better? How do we consider the many possibilities?

We could also consider a model that has an interaction between age and education. `reg <- lm(log.WR ~ edu*age)`

Maybe we don't need the age main effect?
`reg <- lm(log.WR ~ edu*age - age)`

Or perhaps all of the extra education effects are unnecessary?

Modeling is almost as much *art* as science.

notation, again

Cases: $i = 1, \ldots, n$

Target: $y_i \in \mathbb{R}$

Predictors: $X_i = (X_{i1}, \ldots, x_{ip})' \in \mathbb{R}^p$

True model: $Y = \mathbf{X}\beta + \epsilon$

Fitted values: $\hat{Y} = \mathbf{X}\hat{\beta}$,

where $\mathbf{X}$ is the $n \times (p + 1)$ design matrix.

The residuals: $e_i = y_i - \hat{y}_i$

fit: example with Anscombe Data

```
data(anscombe)
head(anscombe)
```

	x1	x2	x3	x4	y1	y2	y3	y4
1	10	10	10	8	8.04	9.14	7.46	6.58
2	8	8	8	8	6.95	8.14	6.77	5.76
3	13	13	13	8	7.58	8.74	12.74	7.71
4	9	9	9	8	8.81	8.77	7.11	8.84
5	11	11	11	8	8.33	9.26	7.81	8.47
6	14	14	14	8	9.96	8.10	8.84	7.04

Anscombe, Francis J. (1973) Graphs in statistical analysis. American Statistician, 27, 17-21

fit: example with Anscombe Data

	x1	x2	x3	x4
mean	9.000	9.000	9.000	9.000
stdev	3.317	3.317	3.317	3.317

	y1	y2	y3	y4
mean	7.501	7.501	7.50	7.501
stdev	2.032	2.032	2.03	2.031

```
with(anscombe, c(cor(x1,y1),cor(x2,y2),cor(x3,y3),cor(x4,y4)))
[1] 0.8164 0.8162 0.8163 0.8165
```

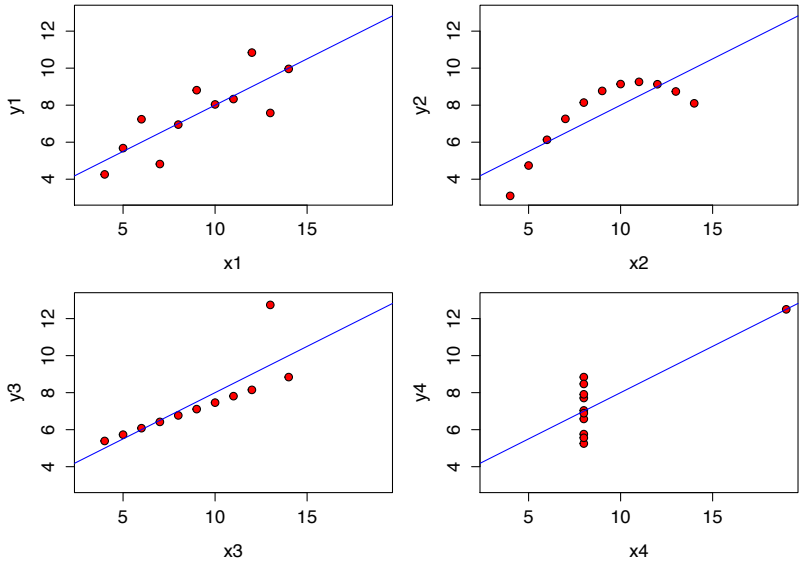
fit: example with Anscombe Data

Four models are fit:

Model		Estimate	$\Pr(>\|t\|)$		Adj R^2	AIC
y1 $\sim$ **x1**	(Intercept)	3.000	0.0257	*	0.629	39.68
	x1	0.500	0.0022	**		
y2 $\sim$ **x2**	(Intercept)	3.001	0.0258	*	0.629	39.69
	x2	0.500	0.0022	**		
y3 $\sim$ **x3**	(Intercept)	3.002	0.0256	*	0.629	39.68
	x3	0.500	0.0022	**		
y4 $\sim$ **x4**	(Intercept)	3.002	0.0256	*	0.63	39.67
	x4	0.500	0.0022	**		

and the statistics all look very similar...

Anscombe's 4 Regression data sets



regression diagnostics

Regression model building is often an iterative and interactive process.

The first model we try may prove to be inadequate.

Regression diagnostics are used to detect problems with the model and suggest improvements.

This is a hands-on process.

residuals

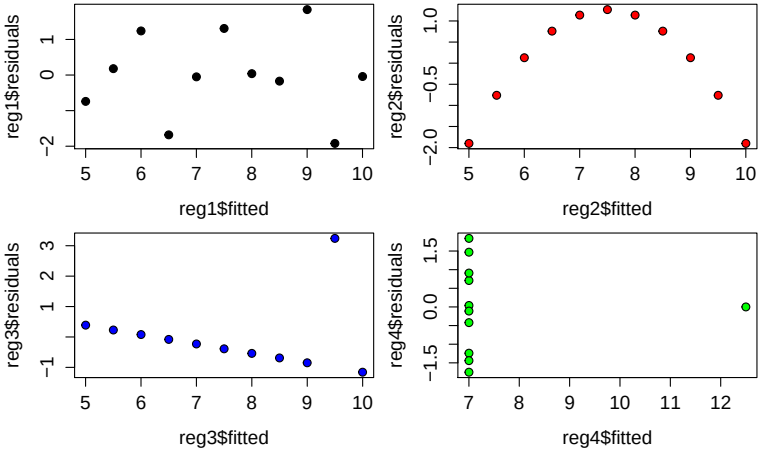
Plotting e_i vs $\hat{y}_i$ is your #1 tool for finding fit problems.

Why?

- Because it gives a quick visual indicator of whether or not the MLR assumptions are true.

fit: example with Anscombe Data

...the statistics were similar but the residuals (plotted against $\hat{y}_i$) are totally different!



residuals

Plotting e_i vs $\hat{y}_i$ is your #1 tool for finding fit problems.

Why?

- Because it gives a quick visual indicator of whether or not the MLR assumptions are true.

What should we expect to see if they are true?

residuals and the model assumptions

Recall that the linear regression model assumes:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij} + \epsilon_i$$

Our goal is to determine if the "true" residuals are iid normal and unrelated to X. If the model assumptions are true, then the residuals must be just noise:

1. Each ϵ_i has the same variance (σ^2).
2. Each ϵ_i has the same mean (0).
3. All of the ϵ_i have the same normal distribution.

Residual plots are useful for checking how well the regression line fits the data, and in particular if there is any systematic lack of fit, for example curvature.

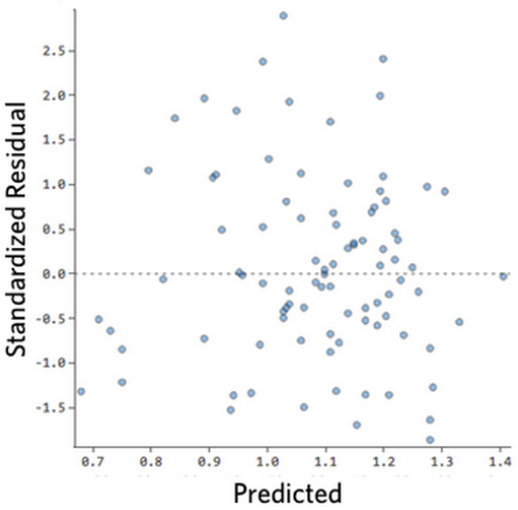
Looking for:

1. points above and below the 0 line
2. no pattern (linear, curved)
3. no change in variation (no “cone”)
4. normal distribution

residual patterns

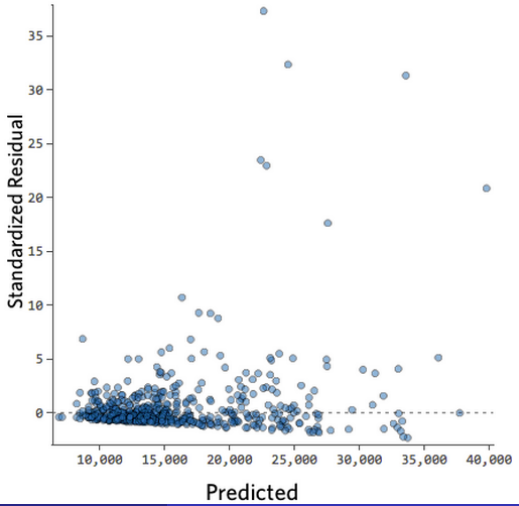
- good residual pattern: none!
- symmetric about 0
- ...maybe a bit too high...

http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression



residual patterns

- heteroscedasticity
- possible fix:
  transform variables
- note: heteroscedasticity does *not* result in biased parameter estimates; but implies OLS is no longer the *best linear unbiased estimator*



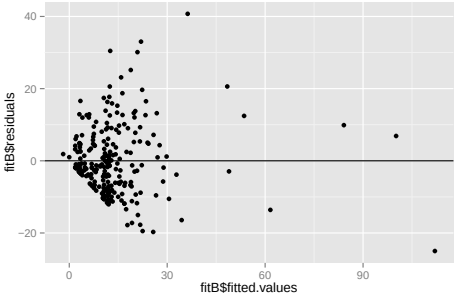
http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression

residual patterns

`ncvTest` in the `car` package

Computes a test of the hypothesis of constant error variance against the alternative that the error variance changes with the fitted values

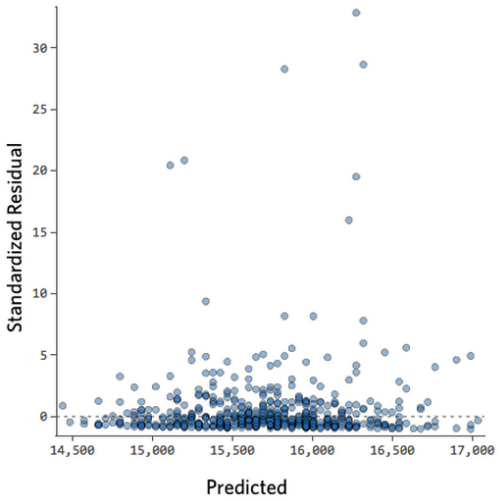
```
Non-constant Variance Score Test
Variance formula:  fitted.values
Chisquare = 46.99 Df = 1 p = 7.152e-12
```



Breusch, T. S. and Pagan, A. R. (1979) A simple test for heteroscedasticity and random coefficient variation. Econometrica 47, 1287-1294.

residual patterns

- unbalanced along residual axis
- very high residuals
- possible fix: transform *outcome* variable

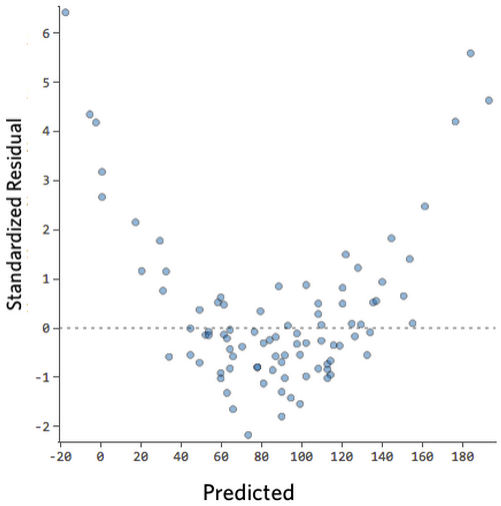


http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression

residual patterns

- non-linear
- possible fix:
  add new variables, e.g.
  quadratic terms

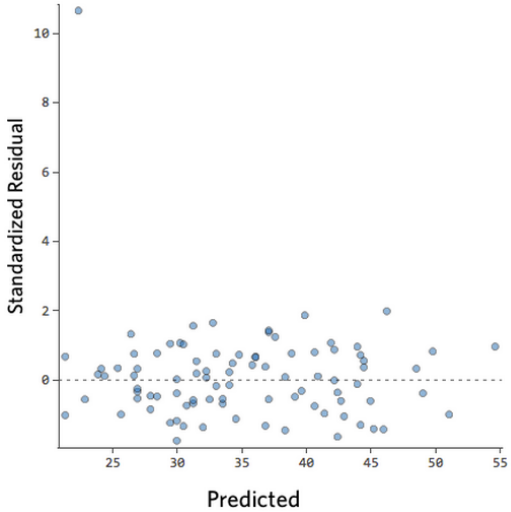
http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression



residual patterns

- residual outlier(s)
- possible fix:
  add variable; fix outlier;
  delete outlier; evaluate
  impact of outlier

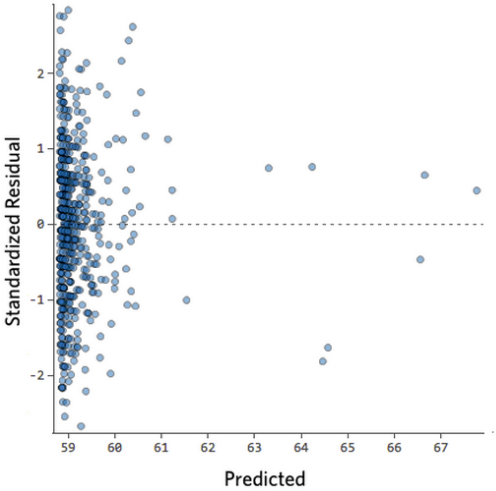
http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression



residual patterns

- unbalanced along predicted axis
- very high residuals
- possible fix: transform *input* variable

http://docs.statwing.com/interpreting-residual-plots-to-improve-your-regression

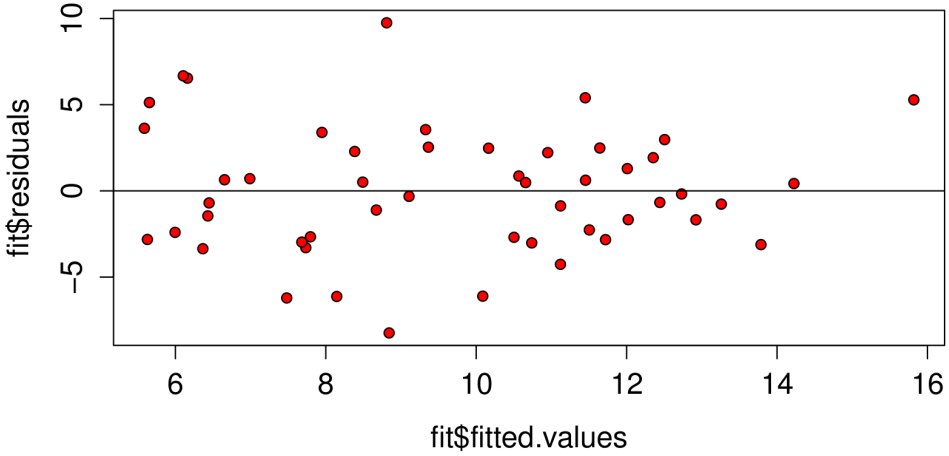


fit example: savings ratio

```
library(datasets)
data(LifeCycleSavings)    # data for 50 countries

# fit a linear model with 4 predictors
fit<-lm(data=LifeCycleSavings,
        sr ~ pop15 + pop75 + dpi + ddpi)
```

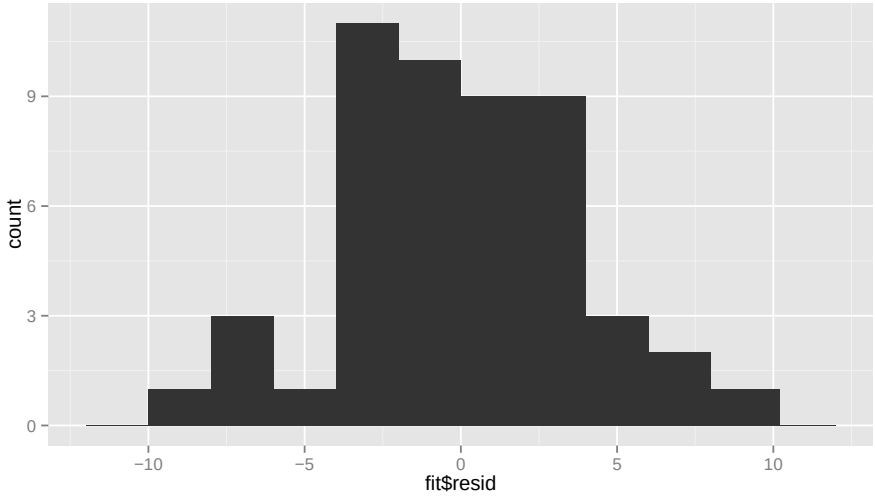
residuals vs. predicted values



test for heteroscedasticity

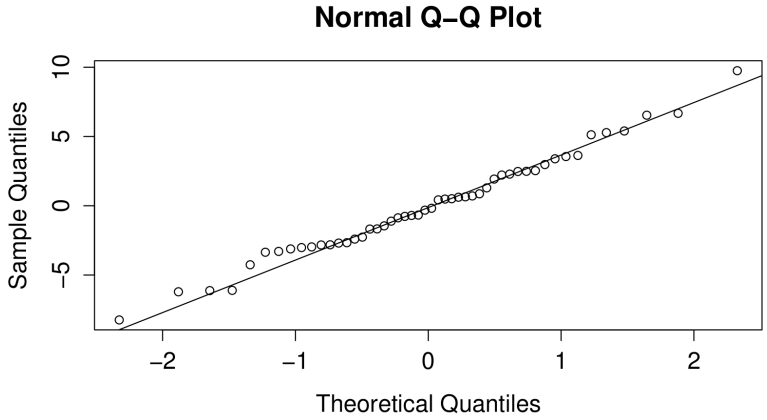
```
> ncvTest(fit)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.274    Df = 1     p = 0.1315
```

histogram of residuals

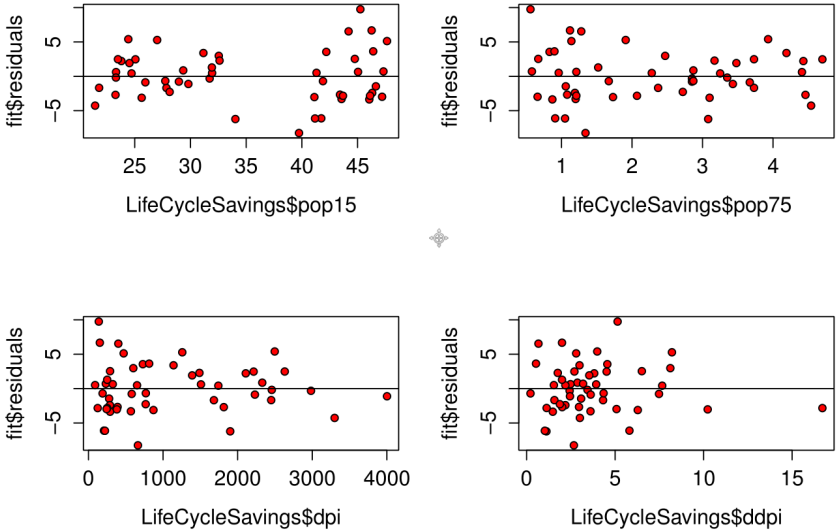


QQ Plot of residuals

```
qqnorm(fit$resid)
qqline(fit$resid)
```



partial residual plots



How do we check these more systematically?

$\longrightarrow$ For example, how "big" is "big" for a residual?

The true ϵ_i errors are unknown, so we look at the least squares estimates of the error.

- $e_i = y_i - \hat{y}_i$ the observed residuals *estimate* the true error.

What should the e_i look like if the MLR model is true?

we will answer this question after introducing a new concept...

hat matrix

Recall:

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y$$

Therefore:

$$\begin{aligned}\hat{Y} &= \mathbf{X}\hat{\beta} \\ &= \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'Y \\ &= \underbrace{\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'}_{\mathbf{H}}Y \\ \hat{Y} &= \mathbf{H}Y\end{aligned}$$

residuals and true error

$$
\begin{aligned}
e &= Y - \mathbf{H}Y \\
&= (\mathbf{I} - \mathbf{H})\, Y \\
&= (\mathbf{I} - \mathbf{H})\, (\mathbf{X}\beta + \epsilon) \\
&= (\mathbf{I} - \mathbf{H})\, \mathbf{X}\beta + (\mathbf{I} - \mathbf{H})\, \epsilon \\
&= (\mathbf{X}\beta - \mathbf{H}\mathbf{X}\beta) + (\mathbf{I} - \mathbf{H})\, \epsilon \\
&= \left(\mathbf{X}\beta - \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{X}\beta\right) + (\mathbf{I} - \mathbf{H})\, \epsilon \\
&= (\mathbf{I} - \mathbf{H})\, \epsilon
\end{aligned}
$$

residuals and true error

How “big” is “big” for a residual?

Problem: e_i retains the scale of the target variable Y.
Solution: standardize by an estimate of the variance of the residual

residuals and true error

$\text{var } \boldsymbol{e} = \text{var}\,(\mathbf{I} - \mathbf{H})\,\epsilon$

since var ϵ is σ^2, then: $\text{var } \boldsymbol{e} = (\mathbf{I} - \mathbf{H})\,\sigma^2$

In scalar form, the estimate for variance of the i^{th} residual is:
$\text{var } e_i = (1 - h_{ii})\,\hat{\sigma}^2$

where h_{ii} are the diagonal elements of $\mathbf{H}$ and
$\sigma^2 \approx \hat{\sigma}^2 = \frac{1}{n-p-1} \sum_{j=1}^{n} e_j^2$

standardized residuals

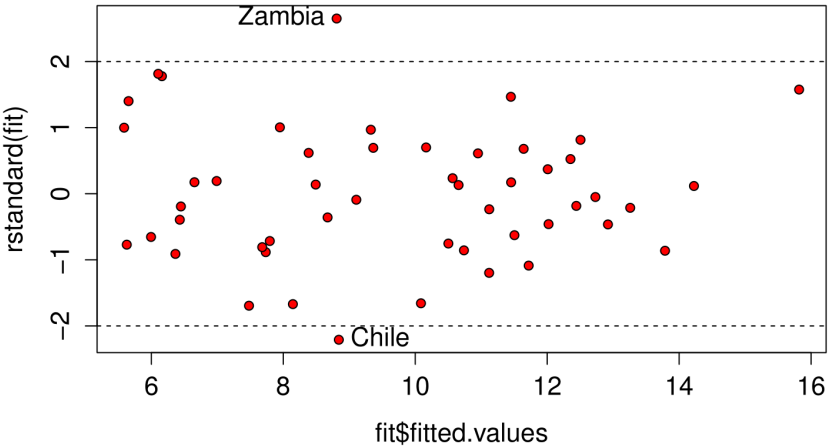
This suggests the use of:

$$r_i = \frac{e_i}{\hat{\sigma}\sqrt{1 - h_{ii}}}$$

to standardize the residuals for equal variance.

Computation in R: `rstandard()`

standardized residuals against predicted values



±2 indicates something unusual; ±3 indicates something really strange;
±4 is something from outer space (it just shouldn't happen)

leverage

Leverage is measured by the so-called "hat-values" from the "hat matrix", **H**.

$\hat{Y} = \mathbf{H}Y$: **H** is an $n \times n$ matrix and projects Y onto the fitted values

Let h_{ij} be the element in the i^{th} and j^{th} column of **H**.

- h_{ij} captures the contribution of y_i to the fitted value $\hat{y}_j$
- diagonal elements $h_i \equiv h_{ii}$ of **H** summarize the leverage of y_i on all fitted values
- the variable Y is not involved in the computation of the hat values
- large values of h_i are due to extreme values in X

- The leverage of an observation measures its ability to move the regression model all by itself by simply moving in the y-direction
  - a point with zero leverage has no effect on the regression model
  - if leverage equals 1, the line must follow the point perfectly
- Range of hat values: $\frac{1}{n} \leq h_i \leq 1$
- Average of hat values: $\bar{h} = \frac{p+1}{n}$
- Rule of thumb: leverage is large if $h_i > \frac{2(p+1)}{n}$
- Issue: for large n, large h_i values are unlikely...
- In R: `hatvalues()`

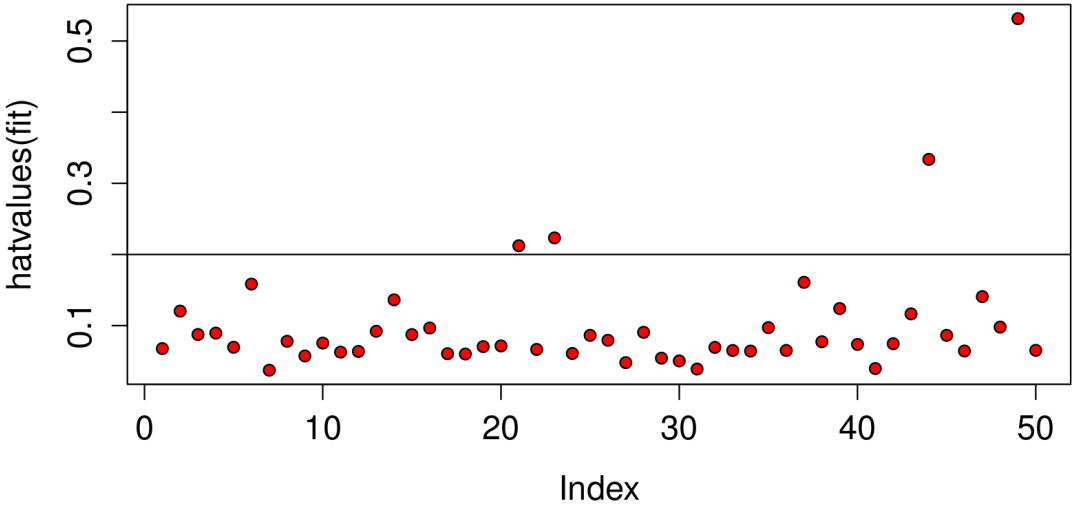
leverage example

```
library(datasets)
data(LifeCycleSavings)    # data for 50 countries

# fit a linear model with 4 predictors
fit<-lm(data=LifeCycleSavings,
       sr ~ pop15 + pop75 + dpi + ddpi)

plot(hatvalues(fit))      # index plot of leverages
abline(abline(h=2*5/50))  # add ref line: 2(p+1)/n

#which obs exceed the rule-of-thumb values?
hatvalues(fit)[hatvalues(fit)>0.2]
```




```
> hatvalues(fit)[hatvalues(fit)>0.2]
```

Ireland	Japan	United States	Libya
0.2122	0.2233	0.3337	0.5315

leverage

Large leverage is not necessarily bad.

Large leverage of large residuals (e.g. multivariate outliers) is possibly very bad.

Back to residuals...

studentized residuals

Problem: The standardized residuals, r_i, still start off with $y_i - \hat{y}_i$ and the problem is that if y_i is really leveraged then it will drag the regression line toward it, influencing the estimate of the residual itself.

Solution: Fit the regression line excluding y_i and base its residual on $y_i - \hat{y}_{(-i)}$, where $\hat{y}_{(-i)}$ denotes the fit based on a regression line estimated excluding y_i.

This idea leads to the studentized residuals.

studentized residuals

Studentized residuals:

- based on “leave one out” idea, (“jackknifing”)
- can be used as a basis for judging the predictive ability of a model. The sum of the squares of the jackknifed residuals is called the PRESS statistic, or Predicted Sum of Squares.
- are an extremely good way of judging how much of an outlier in the y-direction a point is.
- In R: `rstudent()`

outlier tests

Fortunately there is an easy way to compute the studentized residuals which avoids doing n regressions.

$$
\begin{aligned}
t_i &= \frac{e_i}{\hat{\sigma}_{(-i)}\sqrt{(1-h_{(-i)}}} \\
&= r_i \left(\frac{n-p-1}{n-p-r_i^2} \right)^{\frac{1}{2}}
\end{aligned}
$$

And since $t_i \sim t_{n-p-1}$, we can calculate a p-value to test whether case i is an outlier.

outlier tests

We must adjust the level of the test.
Even though it might seem that we only test one or two large t_i values, by identifying them as large we are implicitly testing all cases.

Suppose we want a level α test.

$$\begin{aligned} P(\text{all tests accept}) &= 1 - P(\text{at least one rejects}) \\ &\geq 1 - \sum_i P(\text{test } i \text{ rejects}) = 1 - n\alpha. \end{aligned}$$

This suggests if an overall level α test is required, then a level $\frac{\alpha}{n}$ should be used in each of the tests.

outlier tests

```
library(car)     # load the car library
outlierTest(fit) # uses Bonferroni correction

No Studentized residuals with Bonferonni p < 0.05

Largest |rstudent|:

        rstudent unadjusted p-value Bonferonni p
Zambia     2.854          0.006567       0.3283
```

influence

An influential point is one if removed from the data would significantly change the fit.

An influential point may either be an outlier or have large leverage, or both, but it will tend to have at least one of those properties.

Cook's D

Cook's Distance is a commonly used influence measure that combines these two properties.

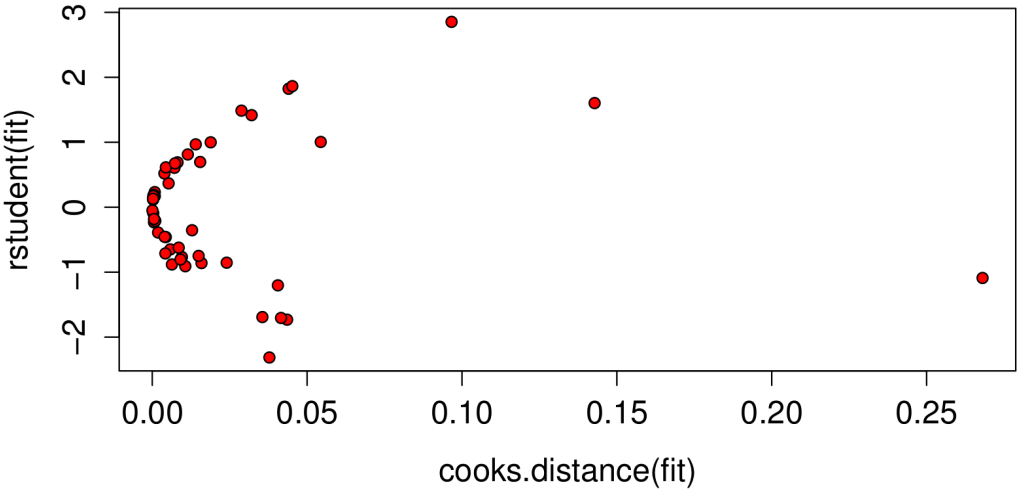
It measures the impact of a given observation on *all* of the model coefficients: how different would they be if observation *i* were deleted?

$$D_i = \frac{e_i^2 \; h_i}{p \; \hat{\sigma}^2 \; (1 - h_i)^2}$$

A value greater than 1 is usually considered influential. Some have suggested: $\frac{4}{n-p-1}$ as a cutoff.

Cooks'D and Studentized Residuals

```
plot(cooks.distance(fit),rstudent(fit))
```



dffits

The dffits statistic measures the influence of an observation on the fitted value for *that observation*.

$$dffits_i = \frac{\hat{y}_i - \hat{y}_{(-i)}}{\hat{\sigma}_{(-i)}\sqrt{h_i}}$$

where $\hat{y}_{(-i)}$ is the estimated value of Y for observation i predicted by the model fit to all the data *except observation* i.

Values of dffits are considered large if they exceed 1, or for large sample sizes if they exceed $2\sqrt{p/n}$.

dfbetas

dfbetas measure the influence of a particular observation on a *particular parameter estimate*. The dfbeta for the effect of observation i on $\hat{\beta}_k$ is calculated as

$$dfbeta_{i,k} = \frac{\hat{\beta}_k - \hat{\beta}_{k(-i)}}{\hat{\sigma}_{(-i)}\sqrt{c_{kk}}}$$

where c_{kk} is the k^{th} diagonal element of the matrix $(\mathbf{X}'\mathbf{X})^{-1}$.

A dfbeta is considered large if it exceeds 1, or for large sample size if it exceeds $2/\sqrt{n}$.

residual, leverage, and influence diagnostics

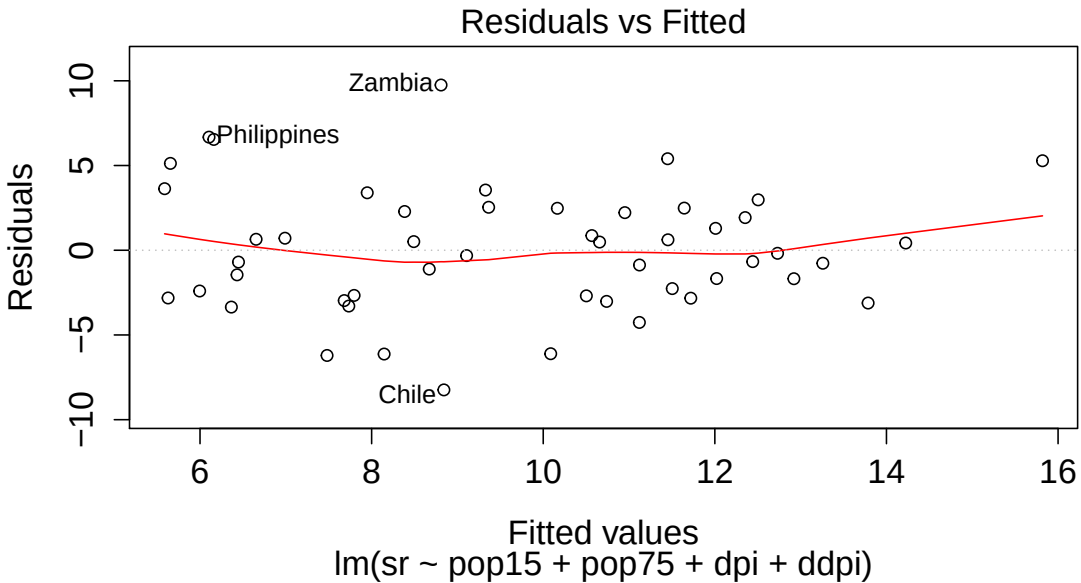
It is helpful to examine these various measures graphically, in two ways:

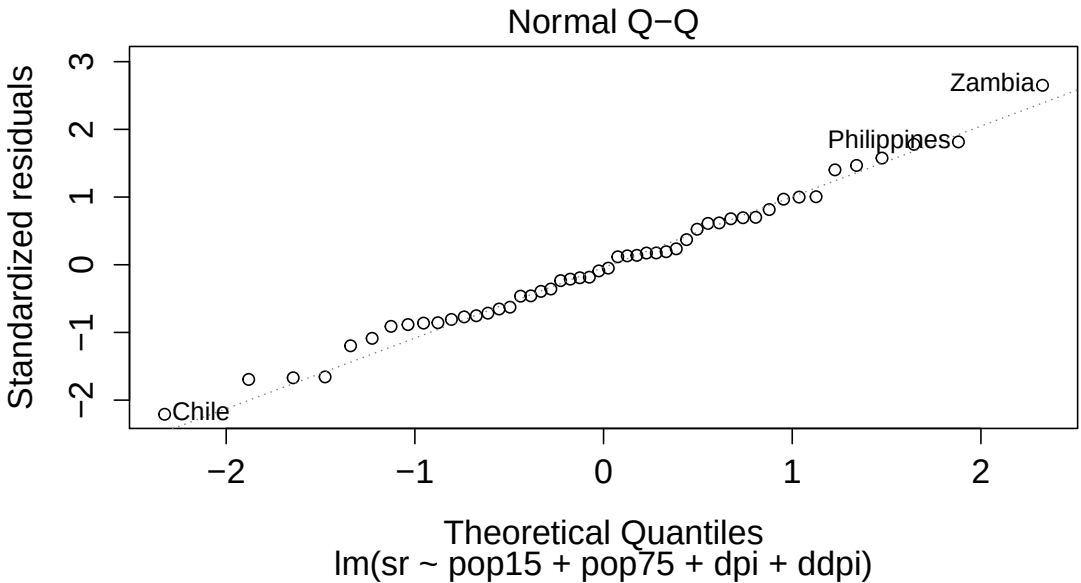
1. look at the distribution of each
2. bivariate scatterplots of any two of the statistics can be examined
   - e.g. Cook's D, dffits, and dfbetas against either the studentized residuals or the leverage values h_i

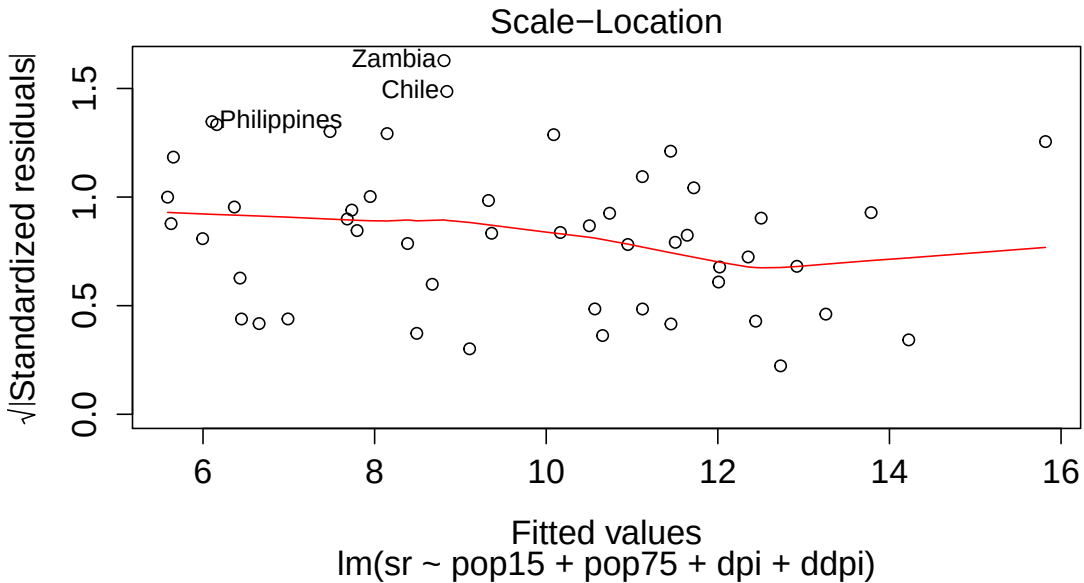
R computes and graphs some of these influence measures for all linear model fits: `plot(fit)`

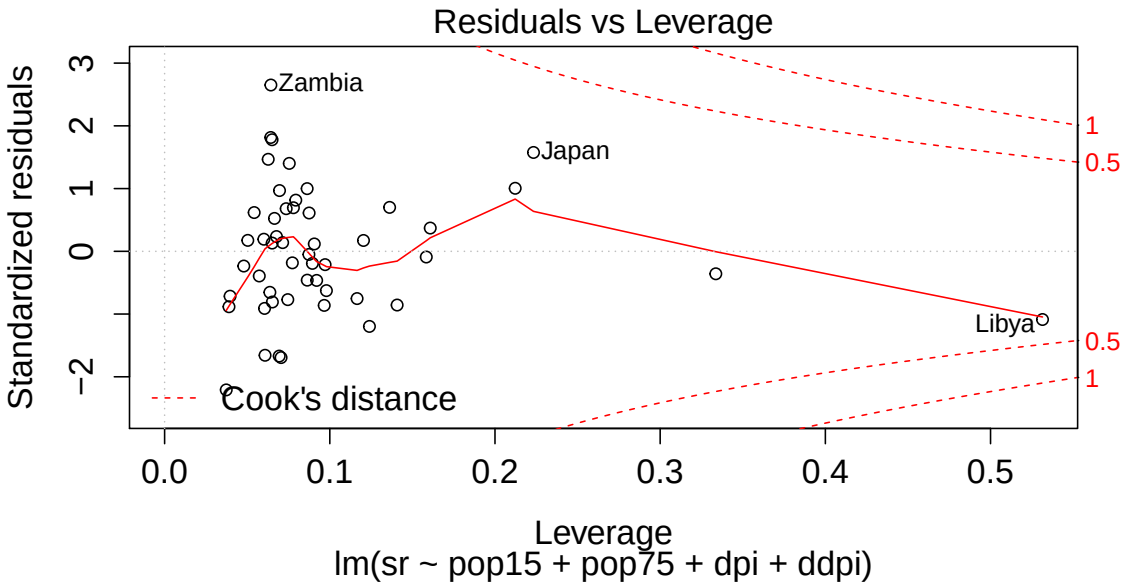
All of these influence measures can be obtained:

`influence.measures()`


Residuals vs Fitted
Zambia
Philippines
Chile
Residuals
10
5
0
−5
−10
6
8
10
12
14
16
Fitted values
lm(sr ~ pop15 + pop75 + dpi + ddpi)


Normal Q−Q
Standardized residuals
3
2
1
0
−1
−2
Zambia
Philippines
Chile
−2
−1
0
1
2
Theoretical Quantiles
lm(sr ~ pop15 + pop75 + dpi + ddpi)

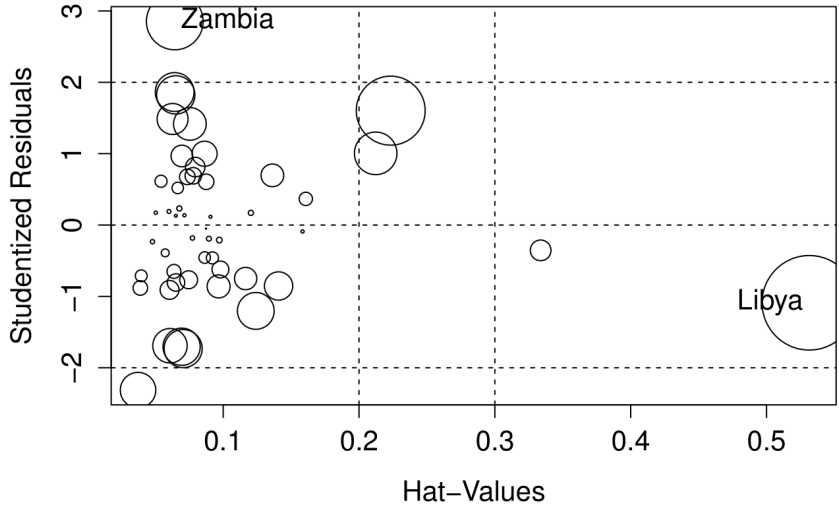

Scale−Location
Zambia
Chile
Philippines
√|Standardized residuals|
0.0
0.5
1.0
1.5
6
8
10
12
14
16
Fitted values
lm(sr ~ pop15 + pop75 + dpi + ddpi)


Residuals vs Leverage
Standardized residuals
3
2
1
0
−2
Zambia
Japan
Libya
1
0.5
0.5
1
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.5
Leverage
lm(sr ~ pop15 + pop75 + dpi + ddpi)

```
> influence.measures(fit)
Influence measures of
 lm(formula = sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings) :
```

	dfb.1	dfb.pp15	dfb.pp75	dffit	cook.d	hat	inf
Australia	0.01232	-0.01044	-0.02653	0.0627	8.04e-04	0.0677	
Austria	-0.01005	0.00594	0.04084	0.0632	8.18e-04	0.1204	
Belgium	-0.06416	0.05150	0.12070	0.1878	7.15e-03	0.0875	
Bolivia	0.00578	-0.01270	-0.02253	-0.0597	7.28e-04	0.0895	
Brazil	0.08973	-0.06163	-0.17907	0.2646	1.40e-02	0.0696	
Canada	0.00541	-0.00675	0.01021	-0.0390	3.11e-04	0.1584	
Chile	-0.19941	0.13265	0.21979	-0.4554	3.78e-02	0.0373	*
China	0.02112	-0.00573	-0.08311	0.2008	8.16e-03	0.0780	
Colombia	0.03910	-0.05226	-0.02464	-0.0960	1.88e-03	0.0573	

The `influencePlot` command in the `car` package:



Size of point $\sim \frac{\sqrt{\text{Cook's D}}}{\max \text{Cook's D}}$

variance inflation factor

Multicollinearity is the condition of two or more of the independent variables being highly correlated.

This includes the situation in which one variable is correlated with some *linear combination* of two or more other variables, even if not correlated with any single variable.

Because of this latter possibility, simple bivariate correlations or scatterplots of the independent variables may not be adequate for detecting collinearity.

variance inflation factor

The most straightforward measure of collinearity adequate to address this situation is called the Variance Inflation Factor (VIF).

There is a VIF for each term in the model. The VIF for the j^{th} term is

$$(VIF)_j = \frac{1}{1 - R_j^2}$$

where R_j^2 is the R^2 for the regression of X_j on all the other X's.

For example, consider our current MLR model from the `LifeCycleSavings` dataset:

$$\text{sr} = \beta_0 + \beta_1\ \text{pop15} + \beta_2\ \text{pop75} + \beta_3\ \text{dpi} + \beta_4\ \text{ddpi}$$

To compute the VIF for variable `pop75`, fit the model:

$$\text{pop75} = \beta_0' + \beta_1'\ \text{pop15} + \beta_2'\ \text{dpi} + \beta_3'\ \text{ddpi}$$

and compute the corresponding R^2 value: 0.84915

$$\textbf{VIF}_{\texttt{pop75}} = \frac{1}{1 - 0.84915} = 6.629$$

variance inflation factor

An individual VIF is considered large – indicative of a problem – if it is larger than 10.

In addition, if the average of the VIFs is considerably larger than 1, this too is considered to indicate a problem.

VIFs do not tell how many collinearities there are, or which variables are included in them.

```
> library(car)
> vif(fit)

pop15 pop75   dpi  ddpi
5.938 6.629 2.884 1.074
```

summary

- evaluation of residuals is key diagnostic approach
- plot standardized or studentized residuals against fitted values
- residual patterns suggest:
  - transforming input and/or output data
  - introducing new variables to the model
- identify residual outliers
- evaluate the leverage, influence associated with residuals
  - hat matrix, Cook's D
  - dffits, dfbetas
- diagnose multicollinearity with VIF

Outline