THE UNIVERSITY OF MEMPHIS

COMP7/8150: Fundamentals of Data Science

Week 8: Statistical Inference/ Single Regression

Max H. Garzon

Standard Methodology

- **Problem Definition/goal**
  - Identify/specify goals of the data analysis
  - commit to specific deliverables
- **Data pre-processing**
  - Identify appropriate data
  - Acquire data (gather, lookup, understand)
- **Data processing**
  - Identify methods (gather, cleanse, store)
  - Carry out the analysis (patterns, trends, predictions?)
- **Data post-processing**
  - Visualize and present
  - Deploy and evaluate. Iterate, if necessary

Learning Objectives

- To refine the concept of statistical inference and how it is applied
- Characterize the assumptions and estimation procedures of a simple linear regression model (RM)
- To identify the various techniques for model checking and diagnostics of RM
- To characterize procedures to perform statistical inferences on the parameters associated with the RM

Linear Regression: Example 1

- R has a *built-in* data frame, called women, with 15 observations on 2 variables. height (inches) x and weight (lbs) y:

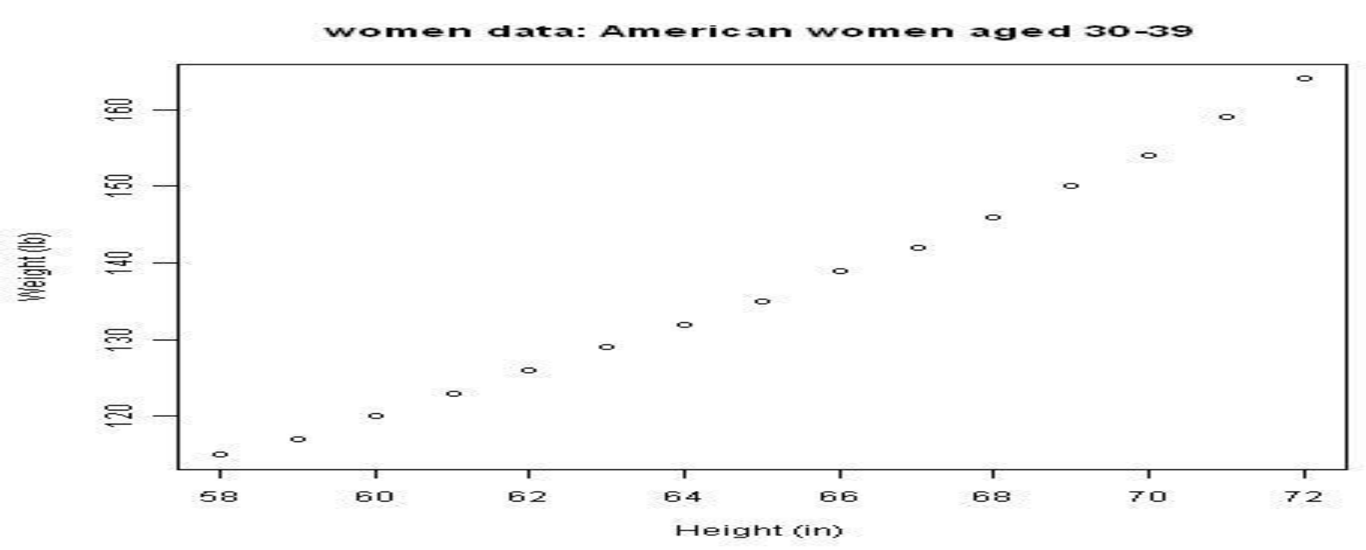
$$y = f(x) + \boldsymbol{\varepsilon}$$

- Need build a model for f(x) that enables us to predict one based on the other.

```
> women
   height weight
1      58    115
2      59    117
3      60    120
4      61    123
5      62    126
6      63    129
7      64    132
8      65    135
9      66    139
10     67    142
11     68    146
12     69    150
13     70    154
14     71    159
15     72    164
```

Plot of women data in R

```
plot(weight~height, data=women, xlab =
"Height (in)", ylab = "Weight (lb)",
main = "women data: American women
aged 30-39")
```



.. Linear Regression: Example 2

- Let y = book weight, which might depend on several variables:

$$y=f(x_1, x_2, x_3, x_4) + \boldsymbol{\varepsilon}$$

x_1 = thickness

x_2 = height

x_3 = width

x_4 = hardback vs. softback indicator

- We want to predict y using knowledge of x_1, x_2, x_3 and x_4 .
- Want to find a metric to evaluate: How good is this prediction?

Simple regression model

- We start with the simplest case, in which the response/target RV *y* is a function of a single independent variable RV *x*.
- How to build a model for the prediction problem of y given x, i.e.,

$$y = f(x) + \boldsymbol{\varepsilon}$$

- Common choice of f(x):
  - $f(x) = \alpha + \beta x$ (linear model)
    $f(x) = \alpha + \beta x + \gamma x^2$ (quadratic model)
    Polynomial models

Steps for fitting simple regression model

- We first plot the data to see the (linear) relationship between x and y
  - In R: `plot(x,y)` or `plot(y~x)`
- If the first order linear model appears to be appropriate, we can estimate parameters for α, β using the formula or functions in R
  - In R: `lm(y~x)`
- Perform a model checking on the model assumptions. In particular, the assumption on the error component
  - In R: `plot(lm(y~x))`
- We can make statistical inferences (confidence interval, or hypothesis testing) for the parameters for α, β
  - In R: `summary()` or `anova()`

A Simple Linear Model

http://www.youtube.com/watch?v=ocGEhiLwDVc

- Data: *n* pairs $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$ from a population or an experiment.
- Model: $y = \alpha + \beta x + \varepsilon$
- Two ways to fit the model, i.e.produce estimates a,b of α, β:

  (1) formula (2) using R

Regression model assumptions

- The regression line $E(y) = \alpha + \beta x$ describes the relationship between the average value of *y* across all values of *x*
- The deviation of *y* from the regression line is denoted by ε
- Usually, we assume ε follows $N(0, \sigma^2)$
- We estimate α and β using the sample via LSE method as described next

Fitting a line to data

Sample: $(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)$

Model:

$$y_i = \alpha + \beta x_i + \epsilon_i, \quad i = 1, 2, \cdots, n,$$

where ϵ_i are i.i.d. with $N(0, \sigma^2)$.

Method of estimation: (LSE)

$$\min_{a,b} \sum_{i=1}^{n} (y_i - a - b x_i)^2$$

Method of Least Squares

- Why is the mean the sum/#?
  In approximating every data point by a single value a, we incur an error for just about every data point x given by |x-a|. Many choices for a
- How can we minimize that TOTAL error?

  $$f(a) = SE(x) = \sum_x (x-a)^2$$

  If we optimize this function using old calculus (take the derivative and set f '(a)=0) the smallest values will be obtained when choosing a = avg of x's = $\sum_x x / n$ where n = # data points x!

.. Method of Least Squares

Method of estimation: (LSE)

$$\min_{a,b} \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

Formula for LSE a (intercept) and b (slope)

$$b = \frac{S_{xy}}{S_x^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}, \quad a = \bar{y} - b\bar{x},$$

where

$$S_{xy} = \frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$$

and

$$S_x^2 = \frac{1}{(n-1)} \sum_{i=1}^{n} (x_i - \bar{x})^2.$$

Computation of $b = \frac{S_{xy}}{S_x^2}$

$$S_x^2 = \frac{1}{(n-1)}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{(n-1)}\left(\sum_{i=1}^{n} x_i^2 - \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n}\right)$$

$$S_{xy} = \frac{1}{(n-1)}\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y}) = \frac{1}{(n-1)}\left(\sum_{i=1}^{n} x_i y_i - \frac{\left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n}\right)$$

Direct computation of **a, b** using R

```
> cov(women$weight, women$height)
[1] 69
> var(women$weight)
[1] 240.2095
> var(women$height)
[1] 20
> s_xy <- cov(women$weight, women$height)
> s_xx <- var(women$height)
> b <- s_xy/s_xx; a <- mean(women$weight)-b*mean(women$height)
> a; b;
[1] -87.51667
[1] 3.45
```

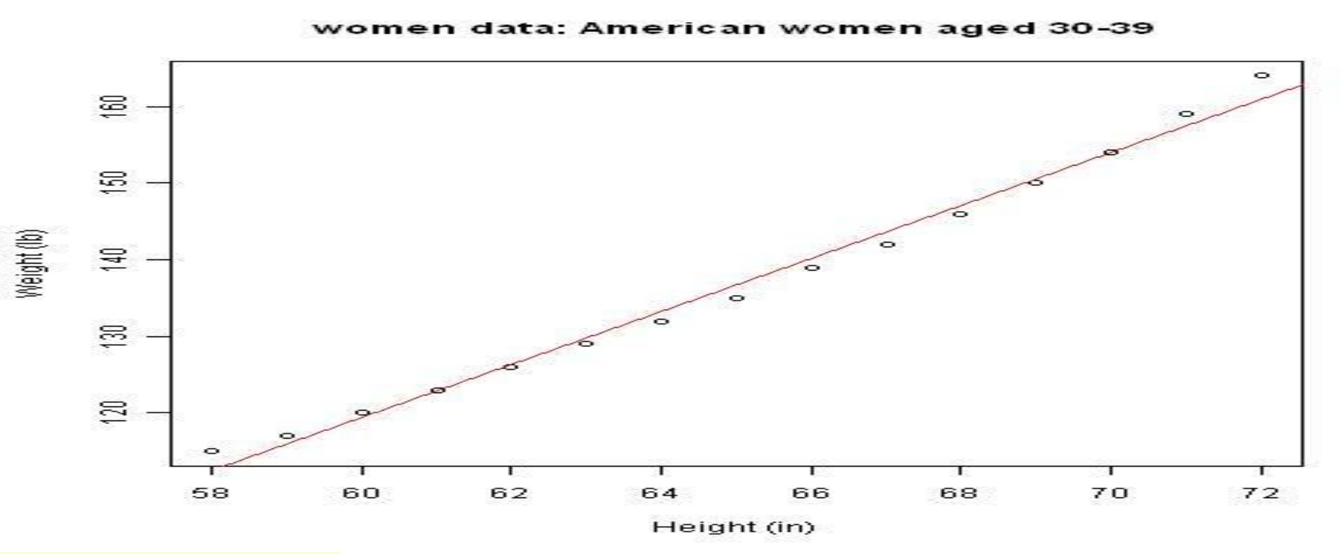
Using lm() in R to compute a, b

```
> lm(weight~height, data=women)

 Call:
   lm(formula = weight ~ height,
data = women)

   Coefficients:
   (Intercept)          height
       -87.52            3.45
```

It is the same as the direct computation

Using lm() to plot regression line

```
plot(weight~height, data=women, xlab =
"Height (in)", ylab = "Weight (lb)", main
= "women data: American women aged 30-39")
abline(lm(weight~height, data=women),
col="red")
```



Fitting linear model using lm()

- Model: $y = \alpha + \beta x + \varepsilon$

  Format in R: lm(y~x)
- Model: $y = \beta x + \varepsilon$

  Format in R: lm(y~ -1+x)
- Model: $y = \alpha + \varepsilon$

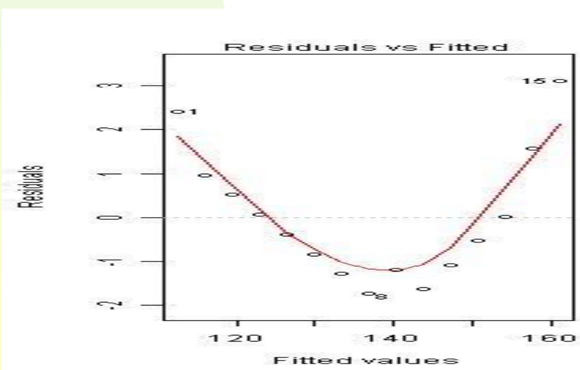
  Format in R: lm(y~1)
- More sophisticated models come when we discuss multiple regression

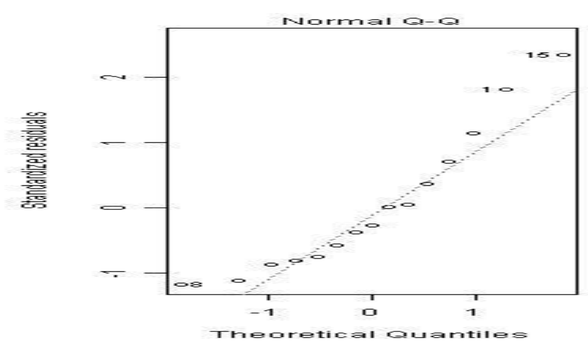
Diagnostic plots on residuals

- A residual = observed value - predicted value

```
> par(mfrow=c(1,2))
> fit <- lm(weight~height, data=women)
> plot(fit, which=1:2)
```

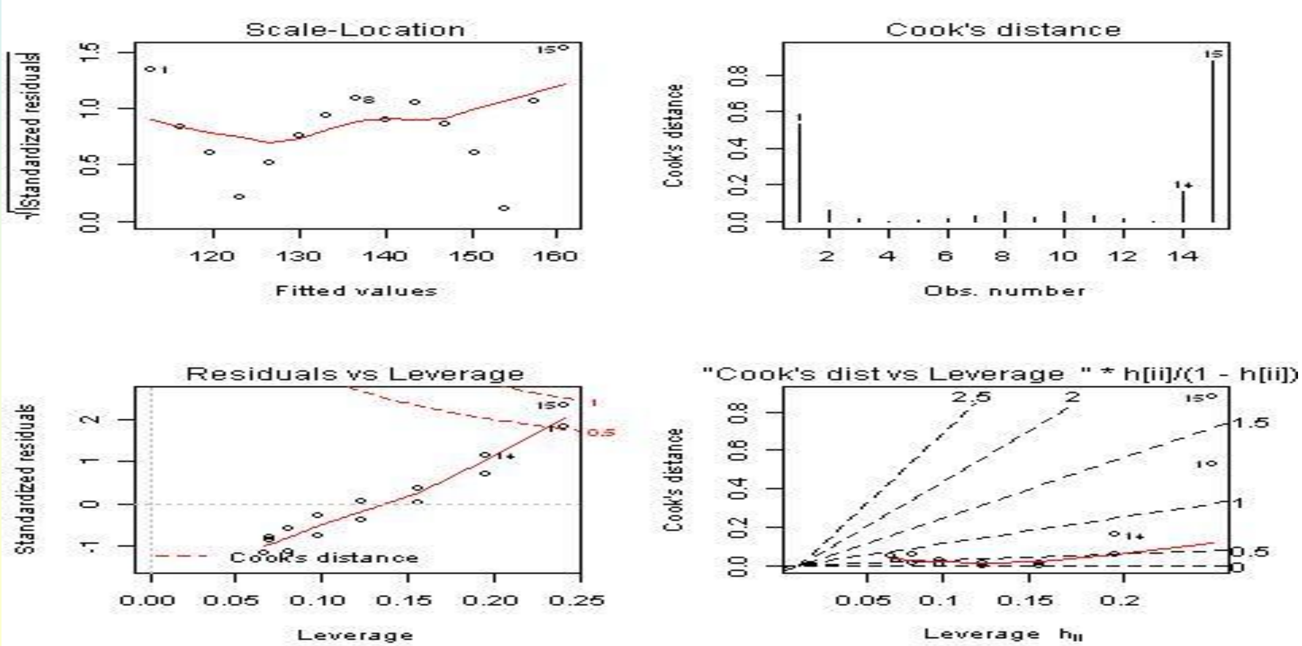
- Left plot: a (quadratic) pattern between fitted and residuals
- Right plot: if the normality assumption is true, the plot should be more like a straight line





Other diagnostic plots

```
> # There are 6 residual plots, their
detailed discussion is beyond the scope
of this class
> par(mfrow=c(2,2))
> plot(fit, which=3:6)
```



Analysis of Variance Table

Total sum of squares SST: $S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$

The Total SS is divided into two parts:

- SSR (sum of squares for regression): measures the variation explained by using *x* in the model (in x significance units).
- SSE (sum of squares for error): measures the leftover variation in y not explained by variation in *x.*

Decomposition of Variation

Total sum of squares is

$$SST = S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

sum of squares for error

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

Total sum of squares due to regression is

$$SSR = SST - SSE$$

Formulas for SST, SSR and SSE

Total sum of squares is

$$SST = S_{yy} = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

sum of squares for error

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} (y_i - a - bx_i)^2$$

Total sum of squares due to regression is

$$SSR = SST - SSE = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 = \frac{S_{xy}^2}{S_{xx}} = \frac{\left(\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})\right)^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

The ANOVA Table

http://www.youtube.com/watch?v=8R6UcK91Cec

Total *df* = n-1
Regression *df* = 1
Error *df* = n-2

Mean Squares

MSR = SSR/(1)

MSE = SSE/(*n*-2)

Source	df	SS	MS	F
Regression	1	SSR	SSR/(1)	MSR/MSE
Error	*n* - 2	SSE	SSE/(*n*-2)	
Total	*n* -1	Total SS		

Building an ANOVA Table in R

- To construct ANOVA table for hypothesis testing, we can use anova() in R.

```
>  fit <- lm(weight~height, data=women)
> anova(fit)
Analysis of Variance Table
Response: weight
                Df  Sum Sq Mean Sq F value    Pr(>F)
height           1 3332.7  3332.7     1433.0 1.091e-14 ***
Residuals 13   30.2     2.3
```

- You should also try to use the formulas given earlier to verify the result.
- More discussion on the outputs from anova() will be given later.

R^2 and Adjusted R^2

- SST = SSR + SSE
- R^2 = SSR/SST is the proportion of the total variation in y that can be explained by using the independent variable x in the model.
- Adjusted R^2 = 1 – [SSE/(n-p-1)]/[SST/(n-1)]

  useful for comparing models with different numbers of parameters (p=1 in this case.)

Computing R^2 and Adjusted R^2

- To find R^2 and Adjusted R^2, we can find it in the output from `summary()` in R.
- More specifically on `summary()` :

```
> fit <- lm(weight~height, data=women)
> fit_summ <- summary(fit)
> names(fit_summ)
  [1] "call"          "terms"          "residuals"
 "coefficients"
  [5] "aliased"       "sigma"          "df"               "r.squared"
  [9] "adj.r.squared" "fstatistic"     "cov.unscaled"
> fit_summ$r.squared
 [1] 0.9910098
> fit_summ$adj.r.squared
 [1] 0.9903183
```

Additional output from lm()

- In addition to finding the regression coefficients (intercept a and slope b), the function lm() in R will return an "object" (say, fit) which can provide more information about the fitting of the linear model.
- We can see the "components" of fit using names(fit):

```
fit <- lm(weight~height, data=women)
> names(fit)
 [1] "coefficients"  "residuals"    "effects"      "rank"
 [5] "fitted.values" "assign"       "qr"           "df.residual"
 [9] "xlevels"       "call"         "terms"        "model"
> fit$coefficients
 (Intercept)      height
   -87.51667     3.45000
```

.. Additional output from lm()

- `> fit <- lm(weight~height, data=women)`
- `> names(fit)`

```
[1] "coefficients"  "residuals"     "effects"       "rank"
[5] "fitted.values" "assign"        "qr"            "df.residual"
[9] "xlevels"       "call"          "terms"         "model“
```

- > fit$fitted.values

```
       1        2        3        4        5        6        7        8
112.5833 116.0333 119.4833 122.9333 126.3833 129.8333 133.2833 136.7333
       9       10       11       12       13       14       15
140.1833 143.6333 147.0833 150.5333 153.9833 157.4333 160.8833
```

Note: fit$fitted.values is the predicted value.

.. Additional output from lm()

- `> fit <- lm(weight~height, data=women)`
- `> names(fit)`

```
[1] "coefficients"  "residuals"     "effects"       "rank"
[5] "fitted.values" "assign"        "qr"            "df.residual"
[9] "xlevels"       "call"          "terms"         "model"
```

- `> fit$residuals`

```
          1           2           3           4           5           6
 2.41666667  0.96666667  0.51666667  0.06666667 -0.38333333 -0.83333333
          7           8           9          10          11          12
-1.28333333 -1.73333333 -1.18333333 -1.63333333 -1.08333333 -0.53333333
         13          14          15
 0.01666667  1.56666667  3.11666667
```

Residual = observed value - predicted value

Summary of output from lm()

In addition to the various components from output produced by lm(), we can use the function in R

- `anova()` to produce the ANOVA table
- `summary()` to provide more detailed summary.

- R output

```
> anova(fit)
Analysis of Variance Table
Response: weight
                 Df Sum Sq Mean Sq F value    Pr(>F)
height            1 3332.7  3332.7  1433.0 1.091e-14 ***
Residuals 13   30.2     2.3
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
   ' ' 1
```

Components of output from summary() and anova()

- Like the function `lm()` (and most R functions), the output "objects" of `anova(lm())` and `summary(lm())` can be saved and their "components" can be retrieved.

```
> names(afit)
[1] "Df"       "Sum Sq"   "Mean Sq" "F value" "Pr(>F)"
> afit$Df
[1]  1 13
> afit[2]
          Sum Sq
height    3332.7
Residuals   30.2
```

.. Components of output from summary() and anova()

```
> sfit <- summary(fit);  names(sfit)
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
> sfit[8:10]
$r.squared
[1] 0.9910098
$adj.r.squared
[1] 0.9903183
$fstatistic
value    numdf    dendf
1433.024    1.000   13.000
```

```
> sfit$sigma
[1] 1.525005
```

SE and CI

- We estimate the intercept (α) and slope (β) of the regression model by estimators "a" and "b".

$$b = \frac{S_{xy}}{S_x^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}, \quad a = \bar{y} - b\bar{x},$$

- We need to find SE(a) and SE(b) for confidence interval construction or hypothesis testing on α, β
- We can find them using (1) formula (complicated, given in next slide) or (2) R function to compute SE(a) and SE(b).

Formula for Standard Error

Parameter	estimator	Var= Standard Error^2
α	$a = \bar{y} - b\bar{x}$	$SE(a)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \frac{\sum_{i=1}^n x_i^2}{n}$
β	$b = \frac{S_{xy}}{S_x^2} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2}$	$SE(b)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2}$
$\alpha + \beta x_0$	$(a + bx_0)$	$\sigma^2 \left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right)$

Predicted variance at $x = x_0$:

$$\sigma^2 \left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right)$$

Estimation of σ²

http://www.youtube.com/watch?v=dJR1WqeBgCg

$$\hat{\sigma}^2 = \frac{SSE}{n-2} = MSE = \frac{\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2}{n-2}$$

Property

$$\frac{(n-2)\hat{\sigma}^2}{\sigma^2} = \frac{SSE}{\sigma^2} \sim \chi^2(n-2).$$

How to find it in R ? (verify!)

```
> summ <- summary(lm(weight~height, data=women))
> summ$sigma
[1] 1.525005
```

Finding Estimators and their SE's in R

```
> summ <- summary(lm(weight~height, data=women))
> names(summ)
 [1] "call"          "terms"         "residuals"     "coefficients"
 [5] "aliased"       "sigma"         "df"            "r.squared"
 [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
> summ_coeff <- summ$coefficients
> summ_coeff
               Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -87.51667  5.9369440 -14.74103 1.711082e-09
height        3.45000  0.0911365  37.85531 1.090973e-14
> summ_coeff[1,1]
[1] -87.51667
> summ_coeff[1,2]
[1] 5.936944
> summ_coeff[2,1]
[1] 3.45
> summ_coeff[2,2]
[1] 0.0911365
```

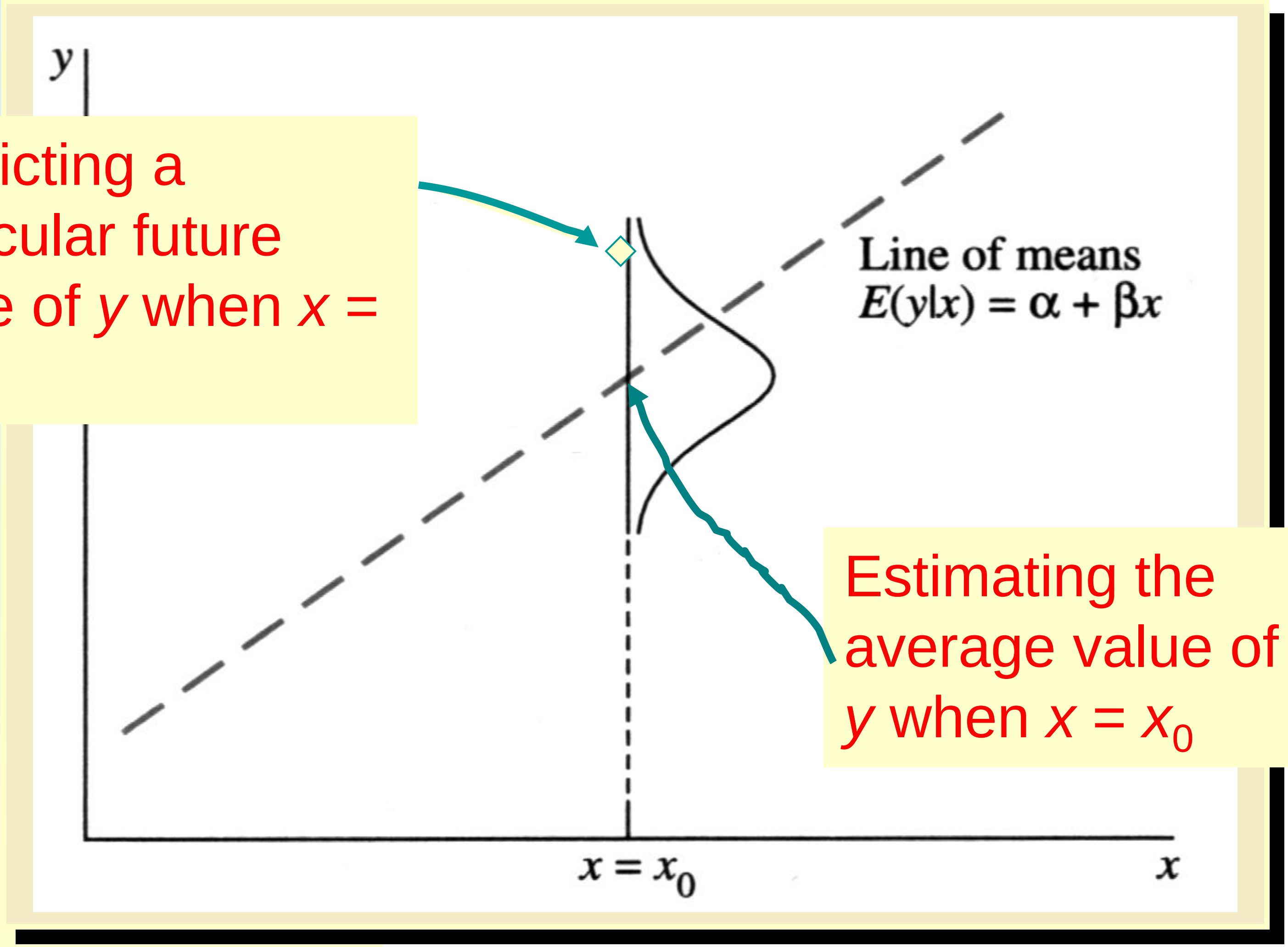
Verify these results with the formula given.

Estimation and Prediction

Given the regression, we can find the confidence interval (CI) for either

- average value of *y* for a given value of x_0
- confidence interval in R using
  > predict(lm(), interval="**confidence**")
  (will produce a narrower interval)
- Predict a future value of *y* for a given x_0.
- prediction interval in R using
  > predict(lm(), interval="**prediction**")
  (will produce a wider interval)

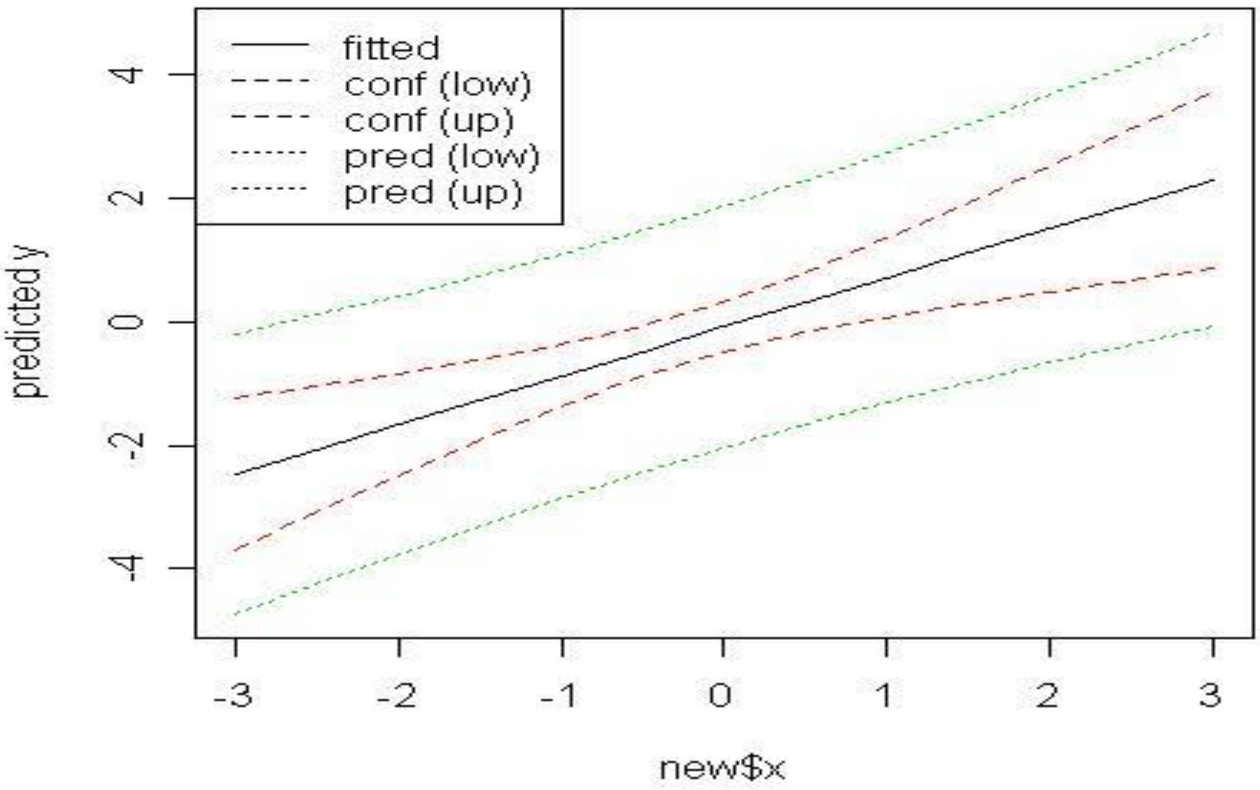
Estimation and Prediction



Predicting a particular future value of y when $x = x_0$

Estimating the average value of y when $x = x_0$

Output plot from R code



Example of using predict() in R

- `x <- rnorm(25); y <- x + rnorm(25)`
- `new <- data.frame(x = seq(-3, 3, 0.5))`
- `fit <- lm(y ~ x)`
- `plim <- predict(fit, new, interval="prediction")`
- `clim <- predict(fit, new, interval="confidence")`
- `matplot(new$x,cbind(clim, plim[,-1]), lty=c(1,2,2,3,3),`
- `col=c(1,2,2,3,3), type="l", ylab="predicted y")`
- `legend("topleft", c("fitted","conf (low)","conf (up)","pred (low)","pred (up)"), lty=c(1,2,2,3,3))`

Constructing Confidence Interval

$$t = \frac{\hat{\theta} - \theta}{SE(\hat{\theta})}$$

How good are a,b for α,β? When sample size is small, the sampling distribution is t(df). Let $t_{\alpha/2}$ = percentile of t(df) distribution, with df=n-2 for simple regression model.

We can construct a 100(1-α)% confidence interval for θ as usual:

$$\hat{\theta} \pm t_{\alpha/2} SE(\hat{\theta})$$

Constructing confidence intervals

```
> fit <- lm(weight~height, data=women)
> confint(fit)
                  2.5 %      97.5 %
   (Intercept) -100.342655 -74.690679
    height        3.253112   3.646888
##verify with the information shown below
                Estimate  Std. Error   t value Pr(>|t|
(Intercept) -87.51667  5.9369440 -14.74103 1.711082e-09
height        3.45000  0.0911365  37.85531 1.090973e-14
```

Testing slope (β)

- Is the model of any value, i.e.,
  is the independent variable *x* of any use in predicting *y*?
- That is, we are testing that the slope of the line b is zero or not.
- H_0: b = 0 vs. H_1: b ≠ 0
- We can use the t-test or, equivalently, use F-test in the anova table.

.. Testing Slope (β) in R

Using lm() in R:

- `> summ <- summary(lm(weight~height, data=women))`
- `># find its components using names(summ)`
- `> summ_coeff <- summ$coefficients`

```
              Estimate     Std. Error   t value      Pr(>|t|)
(Intercept) -87.51667    5.9369440 -14.74103 1.711082e-09
height          3.45000      0.0911365  37.85531 1.090973e-14
```

- Note: t-statistics is 37.85531=3.45/0.0911365.

The F Test

- We can also test the overall usefulness of the model using an F test which is exactly equivalent to the t-test, with $t^2 = F$.

- Using R

```
> anova(fit)
```

Analysis of Variance Table

Response: weight

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
height	1	3332.7	3332.7	1433.0	1.091e-14	***
Residuals	13	30.2	2.3			

(note: F statistic=1433.0=3332.7/2.3= 37.85531^2)

Simulation study on regression model and estimation in R

We would like to simulate data (in R) following a first order linear regression model:

- Generate data for x-variate of n=100 points from certain distribution.
- Choose the parameters for the regression model
  - a <- 2; b <- 1.5; s <- 3
  - y <- a+b*x+e, where e~N(0,s^2).
- With generated data, we then apply the functions lm() and summary() in R to estimate, compare and test about the

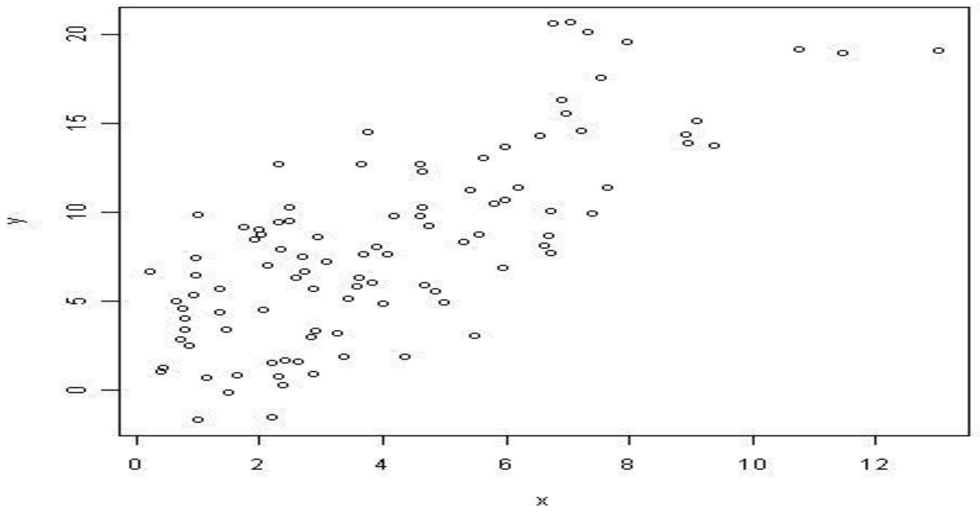
Simulation study on regression model and estimation in R

R code to simulate data following a first order linear regression model.

```
#generate  n=100 points for x from
  chisquare(df=4)
x <- rchisq(100, df=4)
x <- sort(x)
#specify the parameters for the
  regression model
a <- 2; b <- 1.5; s <- 3
e <- rnorm(100,mean=0,sd=s)
y <- a+b*x+e
```

.. Simulation study on regression model and estimation in R

```
#correlation coefficient between x and y
> cor(x,y)
[1] 0.7495822
```



.. Simulation study on regression model and estimation in R

True Model: y=2+1.5x+e, sigma=3

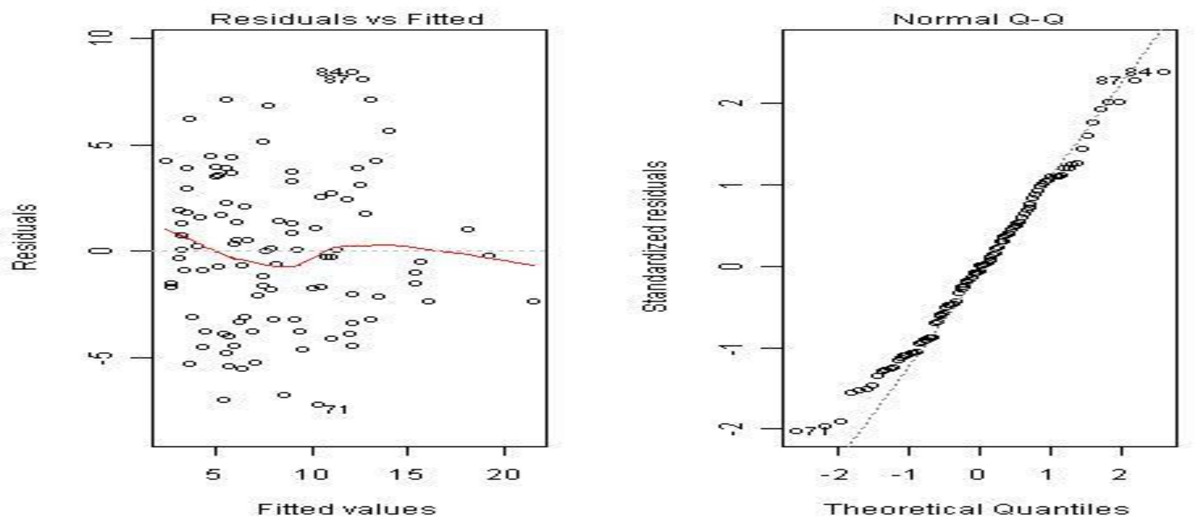
```
> fit <- lm(y~x)
> summary(fit)
…
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   2.0895     0.6481    3.224  0.00172 **
    x         1.4883     0.1328   11.211  < 2e-16 ***

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
Residual standard error: 3.568 on 98 degrees of freedom
Multiple R-squared: 0.5619, Adjusted R-squared: 0.5574
F-statistic: 125.7 on 1 and 98 DF,  p-value: < 2.2e-16
```

.. Simulation study on regression model and estimation in R

- Common diagnostic plots on residuals

```
> fit <- lm(y~x)
> plot(fit, which=1:2)
```

Left plot: checking assumption on homogenous error
Right plot: checking assumption on error normality



Summary: steps for fitting simple regression model

- First plot the data to see ~ linear relationship between x and y
  - In R: `plot(x,y)` or `plot(y~x)`
- If the first order linear model appears to be appropriate, we can estimate parameters for α, β using the formula or functions in R
  - In R: `lm(y~x)`
- Perform a model checking on the model assumptions. In particular, the assumption on the error component
  - In R: `plot(lm(y~x)`
- We can make statistical inferences (confidence interval, or hypothesis testing) for the parameters for α, β
  - In R: `summary()` or `anova()`

Questions?