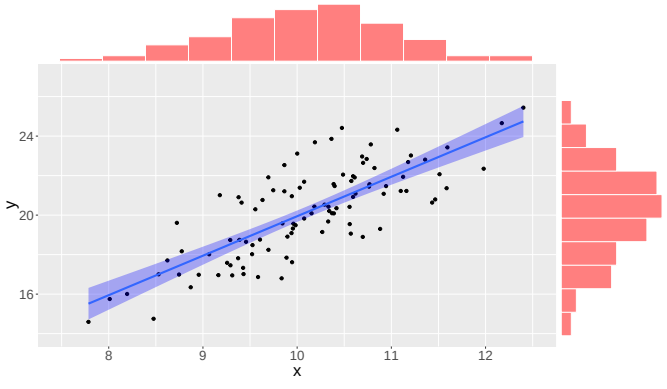
Introduction to OLS with R

Erasmus+ visiting lecture at Faculty of Management & Economics RUB

Athanassios Stavrakoudis

http://stavrakoudis.econ.uoi.gr
Department of Economics, University of Ioannina, Greece

13 January 2016



Contents

What is this about

- An introduction about basic linear models within R.
- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS.
- Download from : www.r-project.org
- R is most conveniently used with **RStudio**, an IDE for R.
- Download from : www.rstudio.com
- There are plenty of books, internet resources and MOOCs about R.
- R the *de-facto* statistical environment in academia. R is heavily used by the industry too.

Packages and libraries

There are several packages used in this lectures. If not present in your system you can install them:

```
if (!require("HH")) install.packages("HH")
if (!require("HistData")) install.packages("HistData")
if (!require("dplyr")) install.packages("dplyr")
if (!require("xtable")) install.packages("xtable")
if (!require("UsingR")) install.packages("UsingR")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("ggExtra")) install.packages("ggExtra")
if (!require("gridExtra")) install.packages("gridExtra")
```

or execute the following script : install.R

RStudio environment

File Edit Code View Plots Session Build Debug Tools Help

Project: (None)

mtcars.R

Source on Save | Run | Source

```
1  rm(list=ls(all=T))
2  data(mtcars)
3
4  plot(mtcars$hp, mtcars$mpg)
5  fit <- lm(mpg ~ hp, data=mtcars)
6  abline(fit, col=4, lwd=2)
7
8  fit <- lm(mtcars$mpg ~ mtcars$hp)
9
10 attach(mtcars)
11 fit <- lm(mpg ~ hp)
12
```

6:26 (Top Level) R Script

Console ~/

```
> rm(list=ls(all=T))
> data(mtcars)
>
> plot(mtcars$hp, mtcars$mpg)
> fit <- lm(mpg ~ hp, data=mtcars)
> abline(fit, col=4, lwd=2)
>
> fit <- lm(mtcars$mpg ~ mtcars$hp)
>
> attach(mtcars)
> fit <- lm(mpg ~ hp)
>
```

Environment History

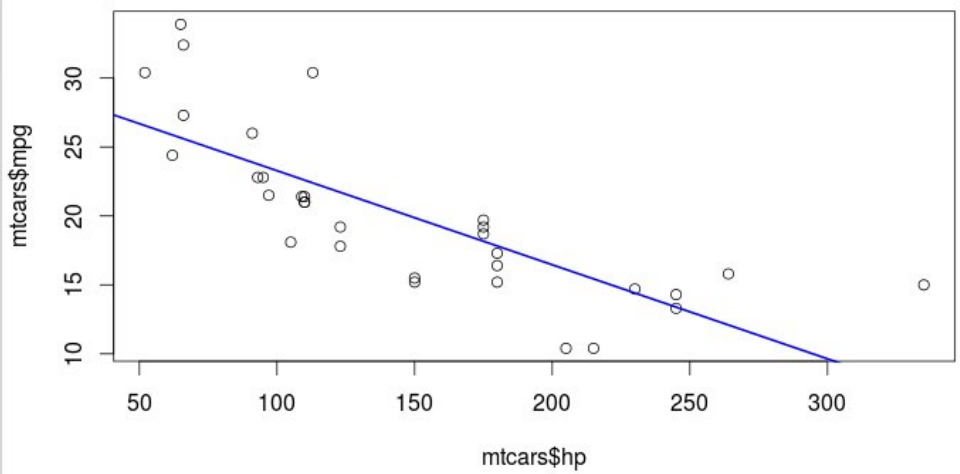
Import Dataset List

Global Environment

Data	
mtcars	32 obs. of 11 variables
Values	
fit	List of 12

Files Plots Packages Help Viewer

Zoom Export Publish



Some introductory commands

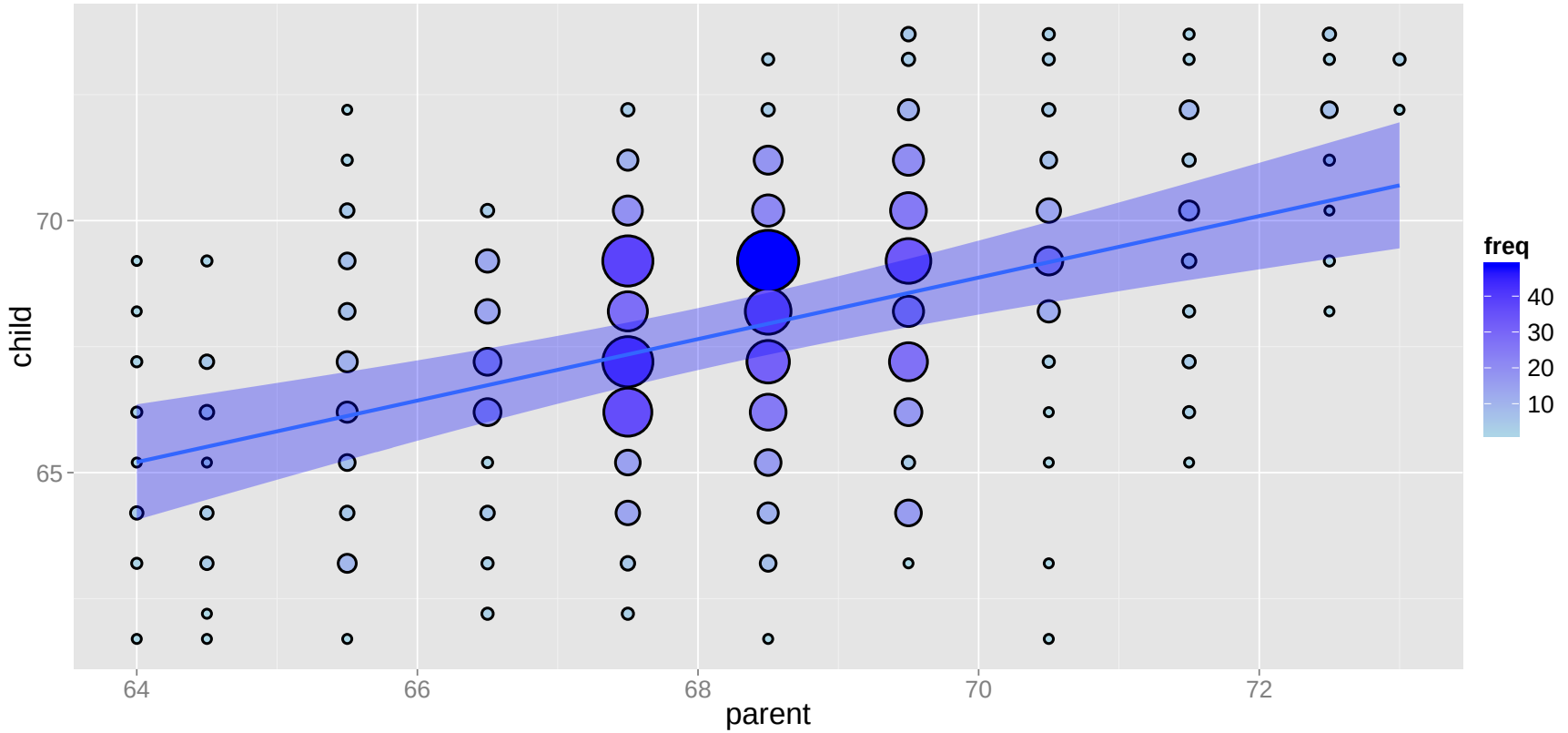
R has lots of methods to load data from local media or network sources. There are many embedded data sets useful for examples.

```
rm(list=ls(all=T))                      # clear the enviroment
data(mtcars)                            # load a data frame

plot(mtcars$hp, mtcars$mpg)             # plot miles per galon
                                        # vs horse power
fit <- lm(mpg ~ hp, data=mtcars)        # fit within the data frame
abline(fit, col=4, lwd=2)               # add regression line

fit <- lm(mtcars$mpg ~ mtcars$hp)       # use variable names

attach(mtcars)
fit <- lm(mpg ~ hp)                     # use attached variables
```

Most of the lecture is about the **lm()** function.

Some history

Over a century ago, F. Galton observed that children's height correlates with their parents height. This is one of the first application of OLS and linear models to science.



F. Galton, Regression towards mediocrity in hereditary stature, *The Journal of the Anthropological Institute of Great Britain and Ireland* **1886**, 15:246–263. doi:10.2307/2841583

Variables in OLS method

$$Y = X\beta + \epsilon$$

Response variable

Response variable (or dependent variable) must be **continuous**.

Predictor variable

Predictor variable(s) (or independent variable) can be:

1. continuous
2. discrete
3. categorical

Data in OLS

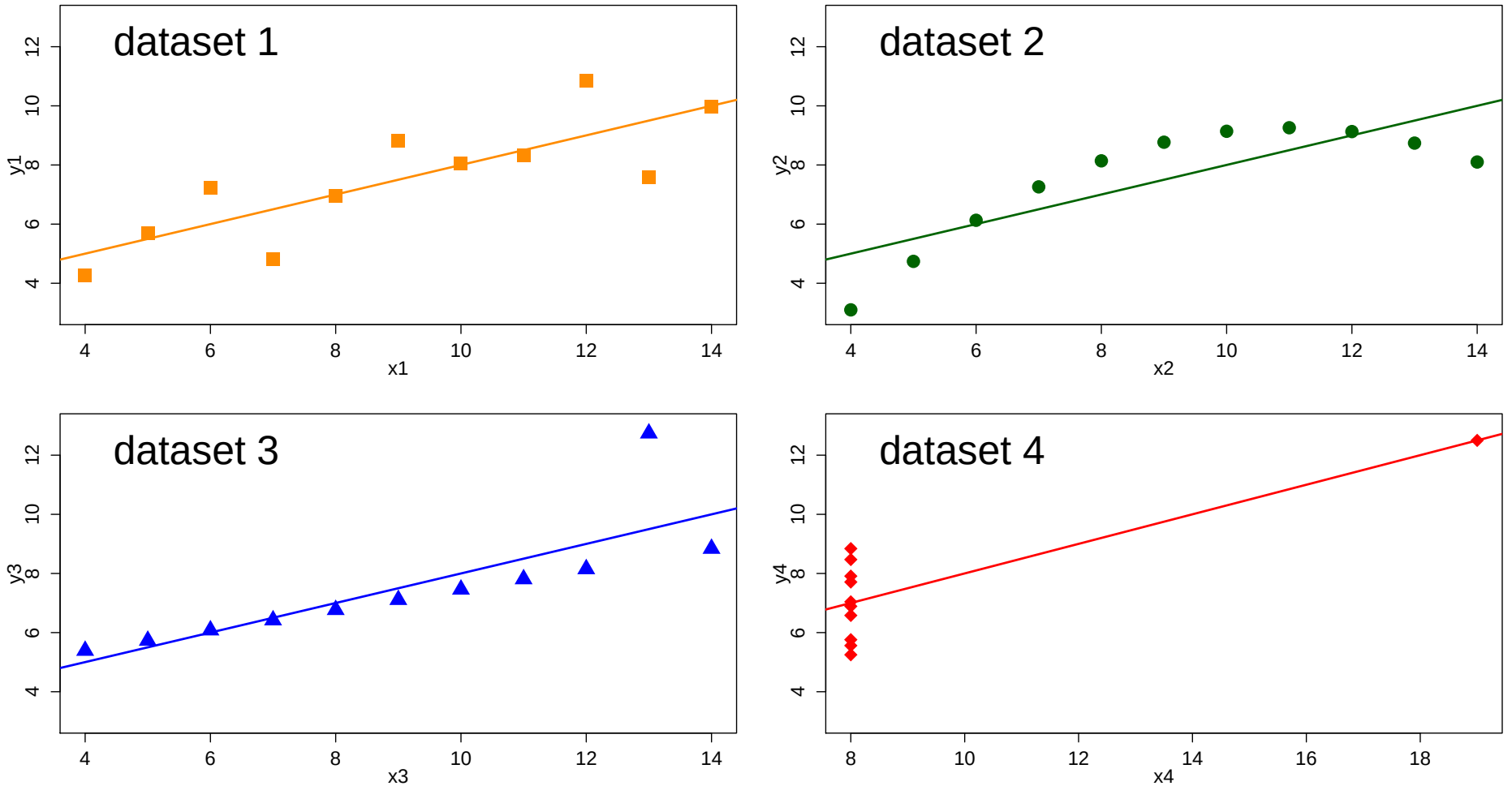
Errors	check/insect the data possible errors
Missing values	Maybe some data values are missed
Patterns	not all data sets are suitable for OLS
Outliers	unusual or extreme values

Plotting

Begin your analysis with plotting of data!

Contents

Anscombe's quartet: four different data sets



F.J. Anscombe, Graphs in Statistical Analysis, *The American Statistician*, **1973**, 27:17–21

Same properties: mean and variance

```
data(anscombe)

> apply(anscombe, 2, mean)
   x1    x2    x3    x4    y1    y2    y3    y4
9.000 9.000 9.000 9.000 7.501 7.501 7.500 7.501

> apply(anscombe, 2, var)
    x1     x2     x3     x4     y1     y2     y3     y4
11.000 11.000 11.000 11.000  4.127  4.128  4.123  4.123
```

Attention!

All four data sets have the same mean and variance (to 2nd decimal point). This does dot mean that come from the same distribution!

Same properties: correlation and OLS coefficients

```
> cor(x1, y1)
[1] 0.8164205
> cor(x2, y2)
[1] 0.8162365
> cor(x3, y3)
[1] 0.8162867
> cor(x4, y4)
[1] 0.8165214
```

```
> coef(lm.1)
(Intercept)          x1
  3.0000909   0.5000909
> coef(lm.2)
(Intercept)          x2
   3.000909    0.500000
> coef(lm.3)
(Intercept)          x3
  3.0024545   0.4997273
> coef(lm.4)
(Intercept)          x4
  3.0017273   0.4999091
```

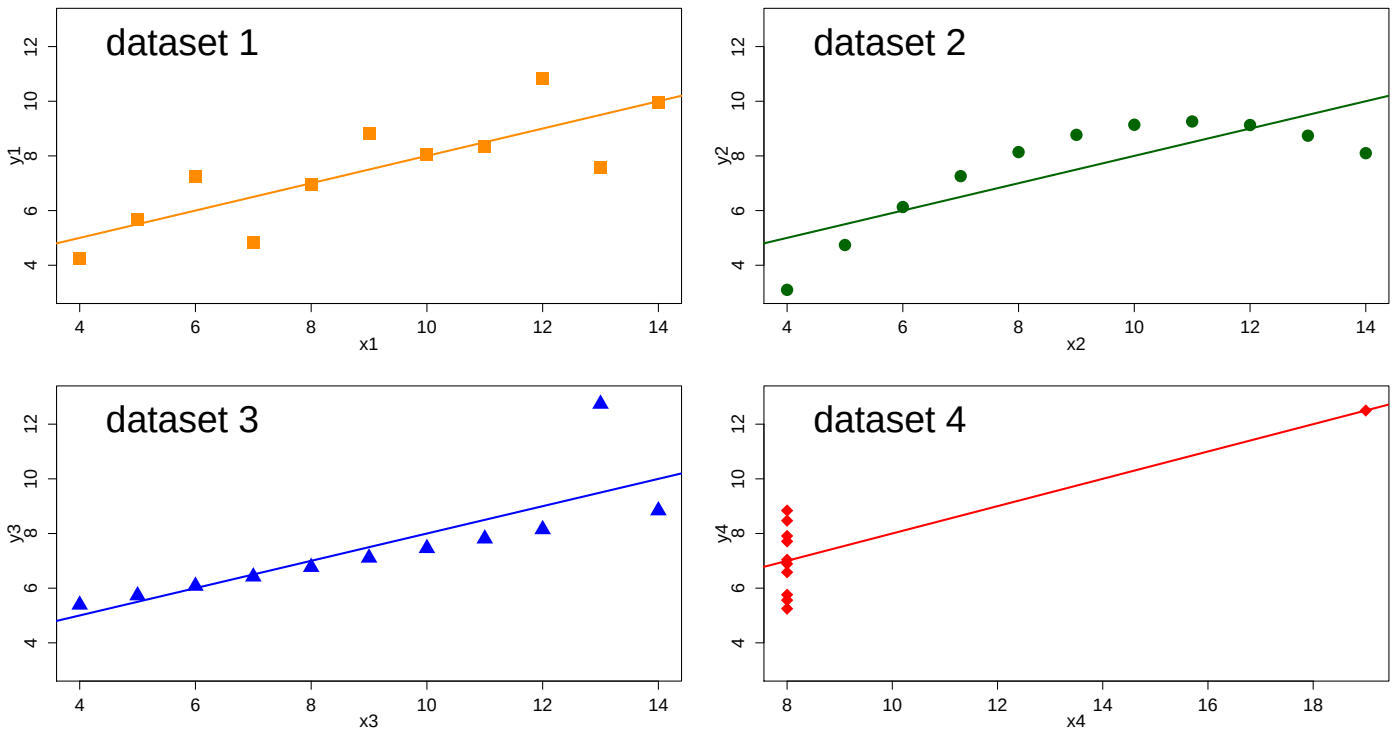
Attention!

Correlation and OLS coeeficients are the same in all four data sets. This does dot mean that imply that in all cases we !

Ancombe's quartet: plot your data

Do not use software as a black box

The four data sets have the same properties but obviously they are different: always plot your data as a first step of your analysis.



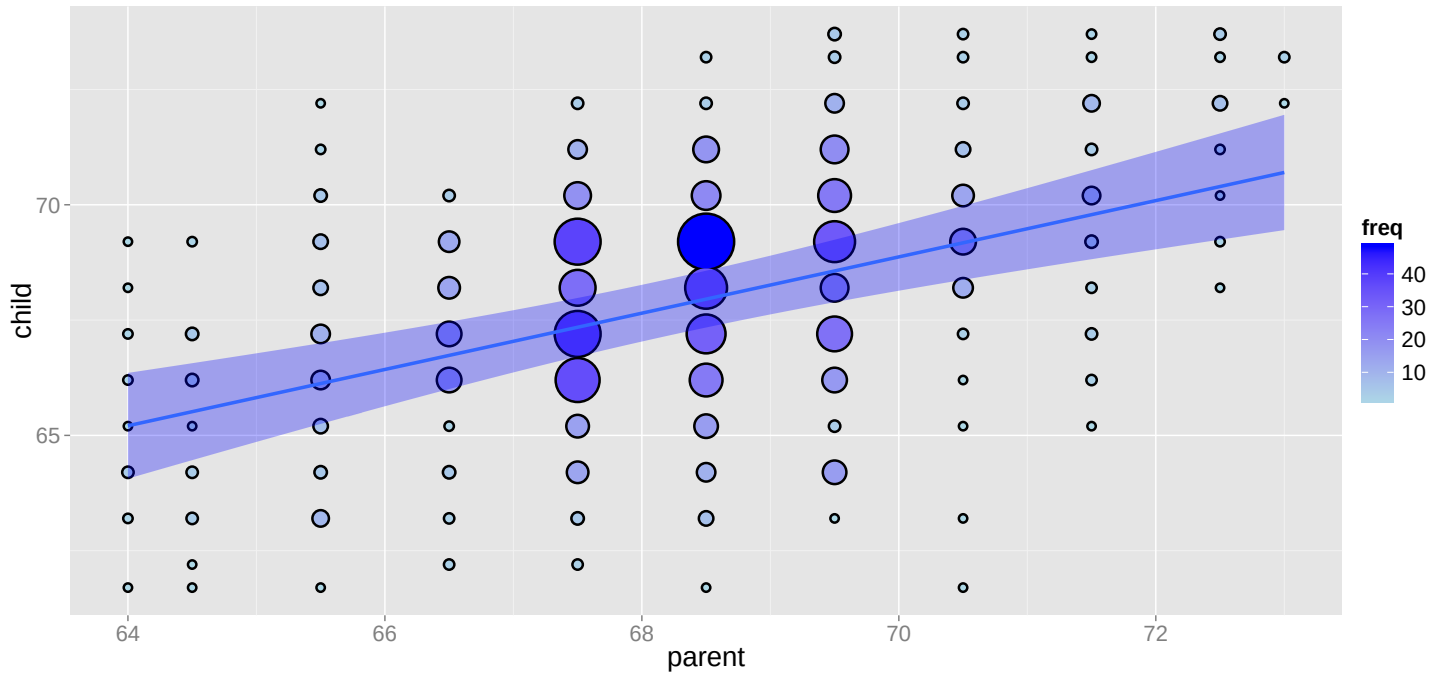
Question

In which case can we reasonably accept the linear model hypothesis?

Contents

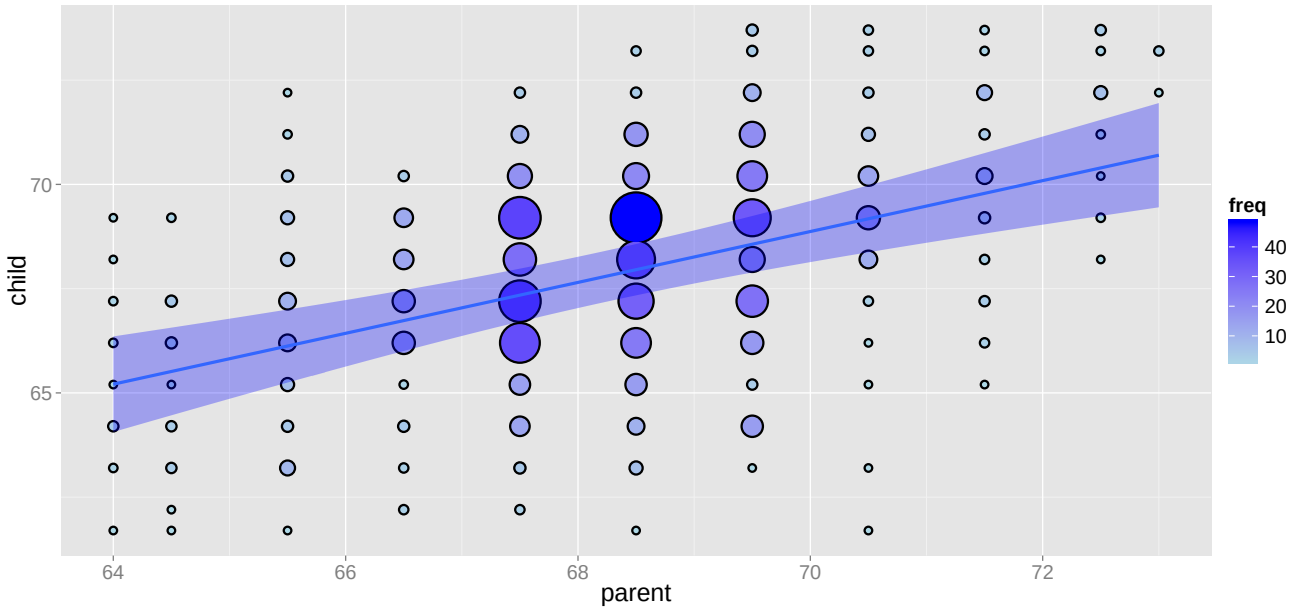
Galton regression

$$\text{childHeight} = \beta_0 + \beta_1\, \text{parentHeight} + \epsilon$$



Galton, F. "Regression towards mediocrity in hereditary stature". *The Journal of the Anthropological Institute of Great Britain and Ireland* **1886**, 15:246–263. doi:10.2307/2841583.

Galton Data



1. A study about parent and child height
2. Units are in inches
3. Parent weight is computed as
   $parent = 0.5 \times (father + 1.08 \times mother)$
4. The graph is overplotted, many points appear with the same coordinates
5. Size of bubble indicates frequency of appearance

Plot Galton data

```
library(HistData)
data(Galton)

plot(Galton$parent, Galton$child)
fit <- lm(child~parent, data=Galton)
abline(fit, col=4, lwd=3)
summary(fit)

library(ggplot2)
ggplot(Galton, aes(x=parent, y=child)) +
  geom_point(shape=5) +
  scale_colour_hue(l=50) +
  stat_smooth(method=lm, se=T, level=0.95,
              fill="blue", alpha=0.3, size=1.2) +
  theme(text=element_text(size=20),
        axis.title=element_text(size=20))
```

Child's height vs parent's height

```
> library(HistData)
> lm (child ~ parent, data=Galton)

Call:
lm(formula = child ~ parent, data = Galton)

Coefficients:
(Intercept)       parent
    23.9415       0.6463
```

Interpretation

1. A child's height (in inches) is $\approx 23.9" + 0.65$ times parent height.
2. A child's height is $\approx 23.9"$ if his/her parent height is $\approx 0"$.
3. If two parents height differ by 1" then their children height will differ by $\approx 0.65"$

A note about variables, lists and numbers

```
> coefficients(fit)          # get the coefficients
(Intercept)      parent
 23.9415302   0.6462906
> fit$coefficients
(Intercept)      parent
 23.9415302   0.6462906
> fit$coef
(Intercept)      parent
 23.9415302   0.6462906
> fit$coef[1]                # the constant term
(Intercept)
   23.94153
> fit$coef[2]                # the slope
   parent
0.6462906
> fit$coef[[2]]              # the slope
[1] 0.6462906
> str(fit$coef[2])
 Named num 0.646
 - attr(*, "names")= chr "parent"
> str(fit$coef[[2]])
 num 0.646
```

Sum of residuals

```
> mean(residuals(fit))
[1] -2.359884e-15
> mean(fit$resid)
[1] -2.359884e-15
> mean(residuals(fit))
[1] -2.359884e-15

> u <- fit$resid
> mean(u)
[1] -2.359884e-15
```

Notes

1. We verify that $\sum \widehat{\epsilon_i} = 0$
2. There are several ways to obtain the result
3. Storing the residuals on a variable (for example `u`) helps if you want further calculations

Getting more results with summary()

```
> summary(fit)

Call:
lm(formula = child ~ parent, data = Galton)

Residuals:
    Min      1Q  Median      3Q     Max
-7.8050 -1.3661  0.0487  1.6339  5.9264

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 23.94153    2.81088   8.517   <2e-16 ***
parent       0.64629    0.04114  15.711   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.239 on 926 degrees of freedom
Multiple R-squared:  0.2105, Adjusted R-squared:  0.2096
F-statistic: 246.8 on 1 and 926 DF,  p-value: < 2.2e-16
```

Coefficients, standard errors and t-statistics

```
> fit$coeff
(Intercept)      parent
 23.9415302   0.6462906

> summary(fit)$coeff
             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 23.9415302 2.81087834  8.517455 6.536845e-17
parent       0.6462906 0.04113588 15.711115 1.732509e-49
```

Summarizing the results

1. Calling **summary()** yields much more detailed results.
2. We do not need only the coefficient values, we also need their Std. Errors to interpret the results.
3. t-statistic and p–values helps us evaluate the significance of the estimation.

Tabulate the results of a regression

```
> library(xtable)
> xtable(fit)
\begin{table}[ht]
\centering
\begin{tabular}{rrrrr}
  \hline
 & Estimate & Std. Error & t value & Pr($>$$|$$t$|$) \\
  \hline
(Intercept) & 23.9415 & 2.8109 & 8.52 & 0.0000 \\
  parent & 0.6463 & 0.0411 & 15.71 & 0.0000 \\
   \hline
\end{tabular}
\end{table}
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	23.9415	2.8109	8.52	0.0000
parent	0.6463	0.0411	15.71	0.0000

Standard error of the coefficient

$$s_{\widehat{\beta}_1} = \sqrt{\frac{\frac{1}{n-2}\sum \hat{\varepsilon}_i^2}{\sum (x_i - \bar{x})^2}}$$

```
> n  <- length(Galton$parent)
> u2 <- fit$resid ** 2
> x2 <- (Galton$parent - mean(Galton$parent))  ** 2
>
> sqrt( sum(u2)/(n-2) / sum(x2) )
[1] 0.04113588
> sqrt(diag(vcov(fit)))
(Intercept)      parent
 2.81087834  0.04113588
```

Standard error of the coefficient

General Model: $Y_i = \beta_0 + \beta_1 \cdot x_i + \epsilon_i$

Galton: $child_i = \beta_0 + \beta_1 \cdot parent_i + \epsilon_i$

$$H_0 : \beta_1 = 0 \quad \longleftrightarrow \quad H_1 : \beta_1 \neq 0$$

T-test statistic:

$$T_1 = \frac{\hat{\beta}_1 - 0}{\hat{\sigma}_{\beta_1}}$$

With normality of ϵ_i under the Null hypothesis:

$$T_1 \sim t_{n-2}$$

p–value: $p = 2 \cdot (1 - F_{t_{n-2}}(|T_1|))$

Obtain p–values analytically

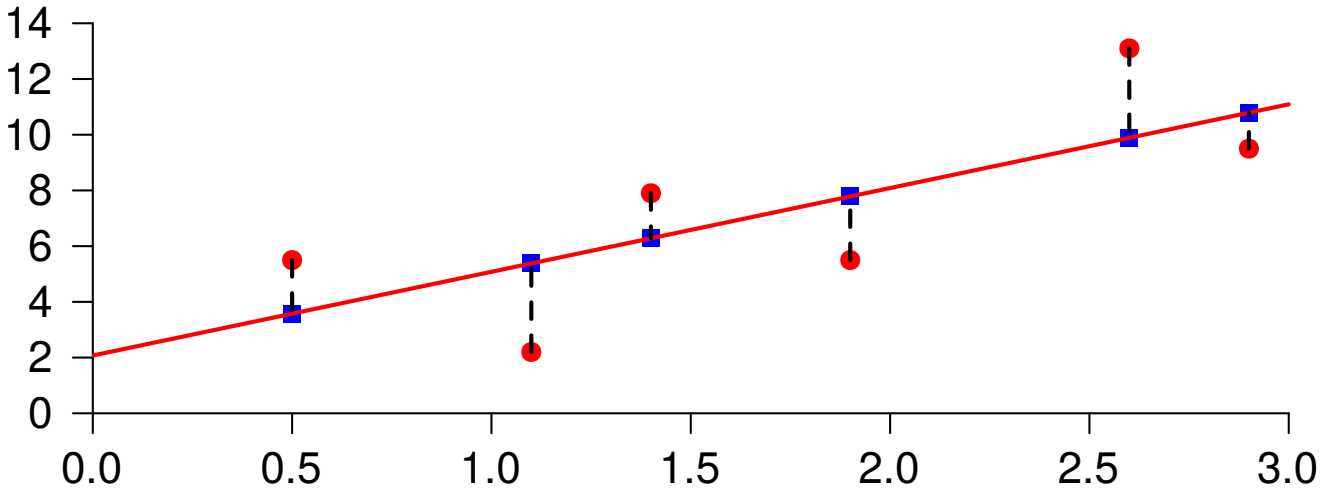
Here is a good example of how to understand and interpret an analytical mathematical formula into R code.

$$p = 2\left(1 - F_{t_{n-2}}(|T_1|)\right)$$

```
> summary(fit)$coeff
             Estimate Std. Error t value  Pr(>|t|)
(Intercept)  23.9415    2.81088   8.517 6.537e-17
parent        0.6463    0.04114  15.711 1.733e-49

n     <- nrow(Galton)
sde   <- sqrt(diag(vcov(fit)))
tstat <- fit$coeff / sde
pval  <- 2 * ( 1 - pt(tstat, df=n-2) )
```

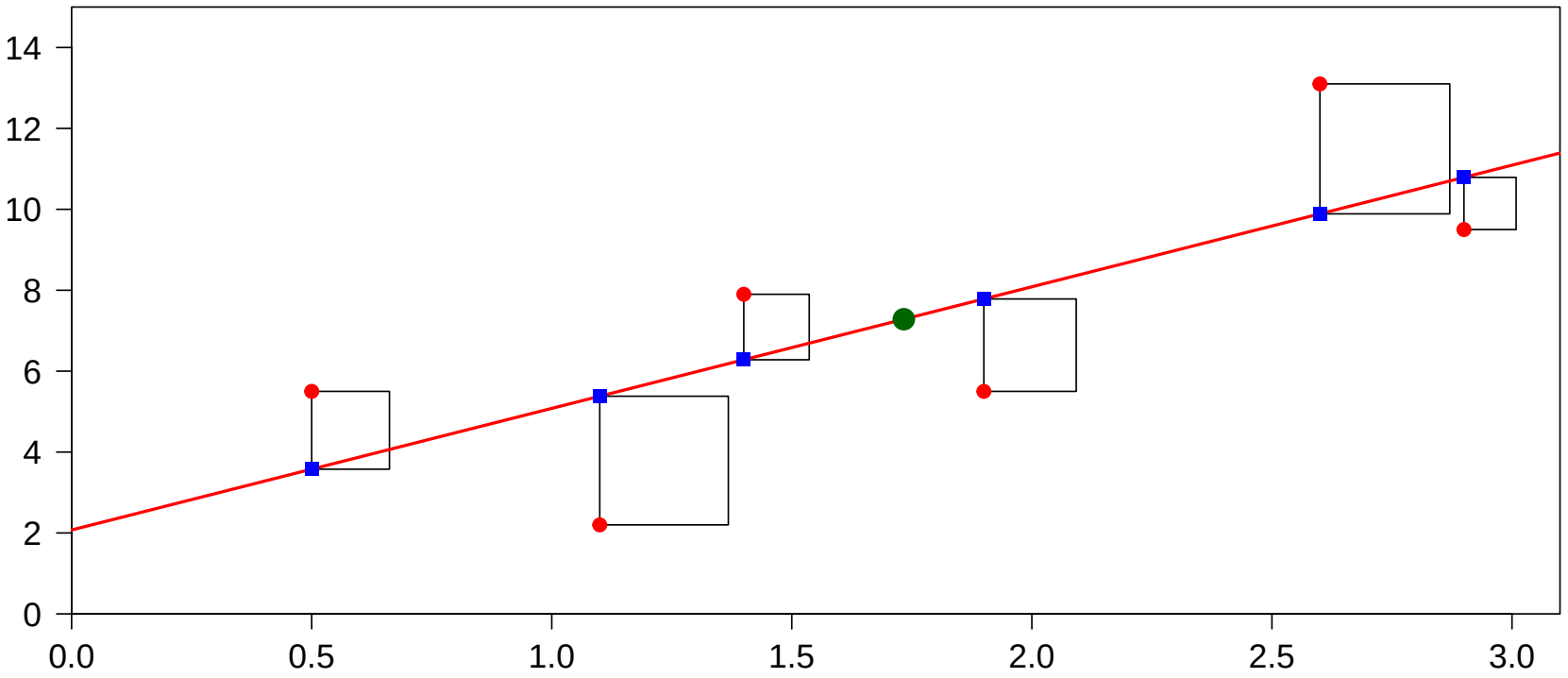
Contents

Basic Terminology



1. **red points**: $\{x_i, y_i\}$, observable variables (price, GDP, temperature, years in work, etc), out of line.
2. **blue points**: $\{x_i, \hat{y}_i\}$, fitted (estimated) data, on the line..
3. **red line**: regression line (fitted from data)
4. **dashed lines**: residuals, $\{\hat{y}_i - y_i\}$ (estimation of errors)
5. **y**: outcome or dependent variable
6. **x**: explanatory variable or independent variable or regressor

Squared residuals



1. **squares**: represent $(\hat{y}_i - y_i)^2$
2. **Regression line** is the one that minimizes:

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

3. **green point**: $(\bar{x}, \bar{y})$, regression line passes through the mean values.

OLS, some reminders

$$
\begin{aligned}
\widehat{Y_i} &= \beta_0 + \beta_1 \widehat{X_i} + \epsilon_i \\
E(\epsilon_i) &= 0 \quad \forall i \quad \text{mean value of residuals} \\
V(\epsilon_i) &= \sigma^2 \quad \forall i \quad \text{homoscedasticity} \\
Cov(\epsilon_i, \epsilon_j) &= 0 \quad \forall i \neq j \quad \text{non correlated residuals} \\
\widehat{\beta_1} &= \text{cor}(Y, x) \frac{\sigma_Y}{\sigma_x} \quad \text{slope of the regression line} \\
\widehat{\beta_0} &= \overline{Y} - \widehat{\beta_1}\, \bar{x} \quad \text{constant term} \\
\widehat{\beta_1} &= \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2} \quad \text{if } \beta_0 = 0
\end{aligned}
$$

Simple linear regression model, estimated in R

Regression equation

Regression equation $\mathbf{y} = \mathbf{X}\beta + \epsilon$

```
lm(y ~ x)
```

Fit the model and estimation of the coefficients

```
Formula <- y ~ x
lm(Formula)
```

Alternative way to call **lm** function

```
fit <- lm(y ~ x)
summary(fit)
```

Store the model to a variable for further usage

```
lm(y ~ x, data=DF)
lm(DF$y ~ DF$x)
```

Variables from a data frame (DF)

```
lm(y ~ x - 1)
lm(y ~ 0 + x)
```

Exclude constant term, two equivalent methods

Estimation in R: quick reference

```
fit <- lm(y ~ x)
```
Store the results of the estimation to a variable

```
summary(fit)
```
Examine the results

```
1 plot(x, y)
2 abline(fit)
```
Quick scatter plot of the data with the regression line

```
coefficients(fit)
```
List the coefficients

```
residuals(fit)
```
Extract the residuals ($\widehat{\epsilon_i}$)

```
fitted.values(fit)
```
Extract the fitted values ($\widehat{y_i}$)

```
confint(fit)
```
Confidence interval

```
vcov(fit)
```
Variance/covariance matrix

Contents

Coefficient of determination R^2

The coefficient of determination (**Multiple R-squared** in R) indicates the proportion of the variance in Y, which is explained by the model.

$$R^2 = \frac{\sum \left(\widehat{Y}_i - \overline{Y}\right)^2}{\sum \left(Y_i - \overline{Y}\right)^2} = \frac{ESS}{TSS} = \frac{S_{xY}^2}{S_x^2 S_Y^2} = r_{xY}^2$$

$R^2 \to 1$ indicates a good fit $R^2 \to 0$ indicates a poor fit

$$TSS = \sum (y_i - \overline{y})^2$$ Total sum of squares

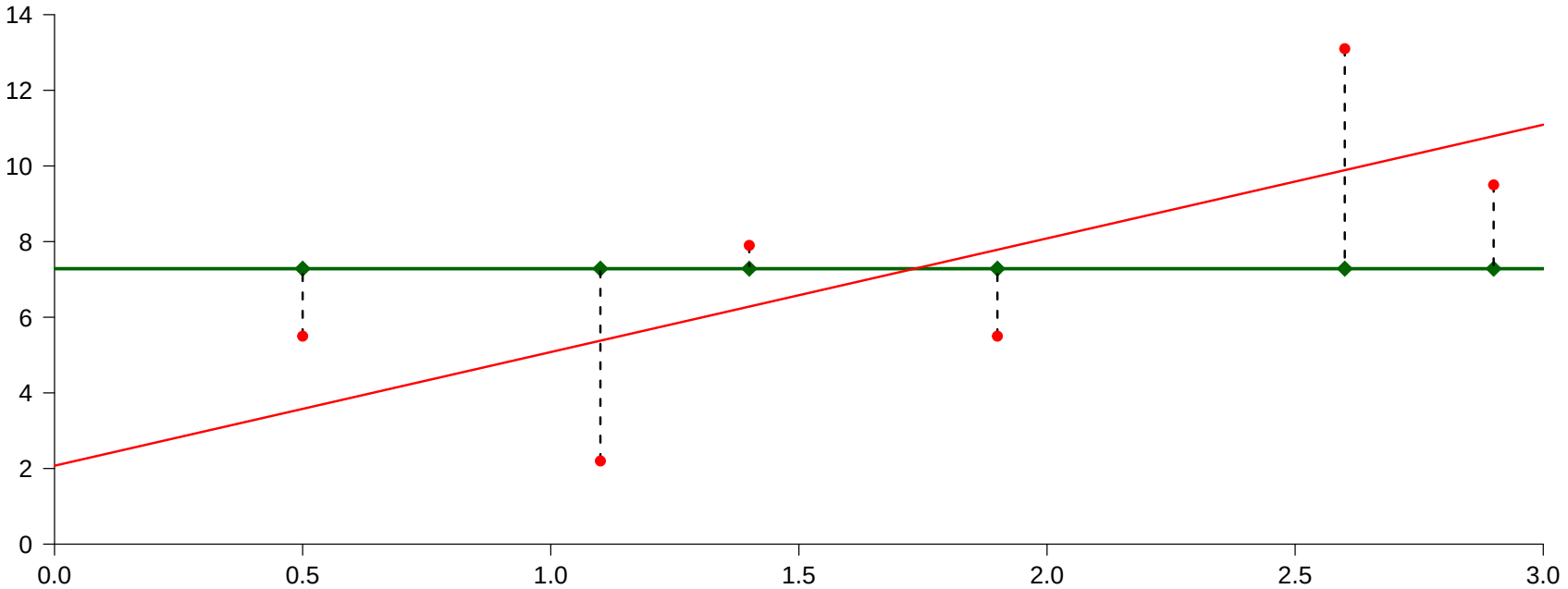
$$ESS = \sum (y_i - \widehat{y})^2$$ Error sum of squares

$$RSS = \sum (\widehat{y}_i - \overline{y})^2$$ regression sum of squares

$$TSS = RSS + ESS$$

$$R^2 = RSS/TSS$$ Fraction of unexplained variance (FUV)

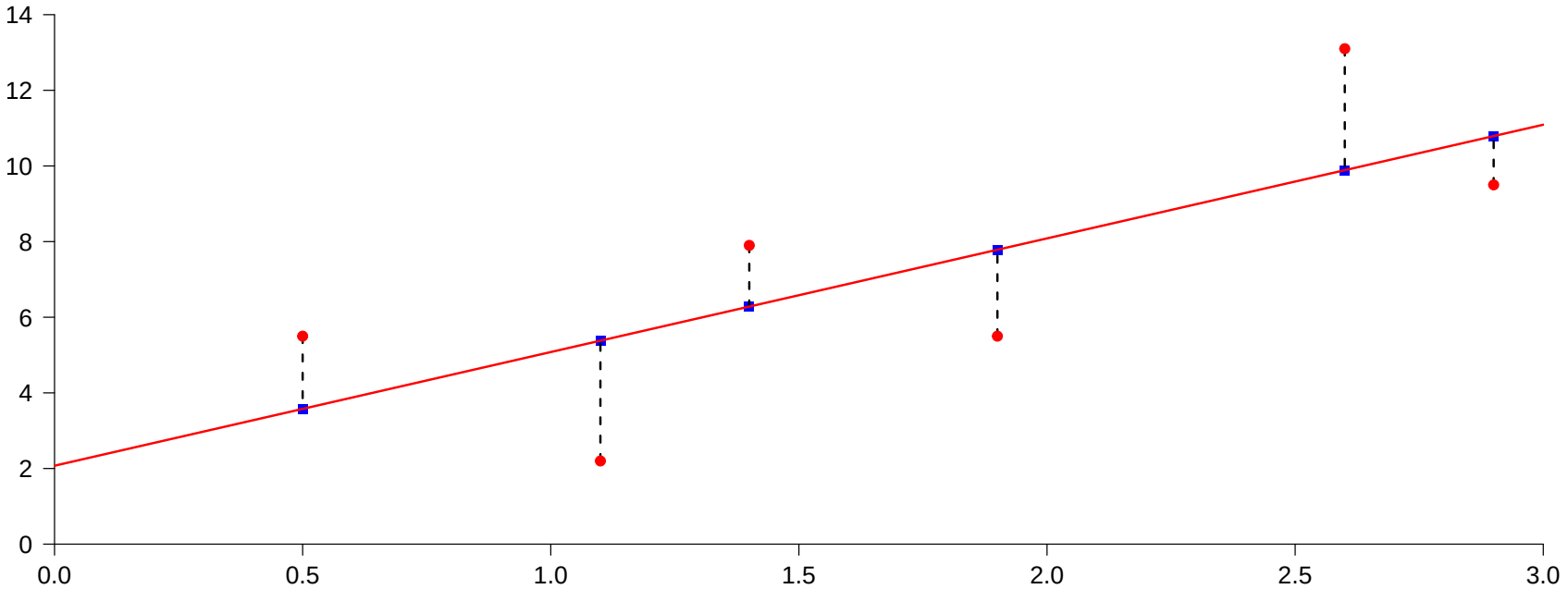
Total sum of squares (TSS)



$$TSS = \sum (y_i - \overline{y})^2$$

1. **red points**: y_i, observations
2. **green points**: $\overline{y}$, mean value
3. **dashed lines**: $y_i - \overline{y_i}$

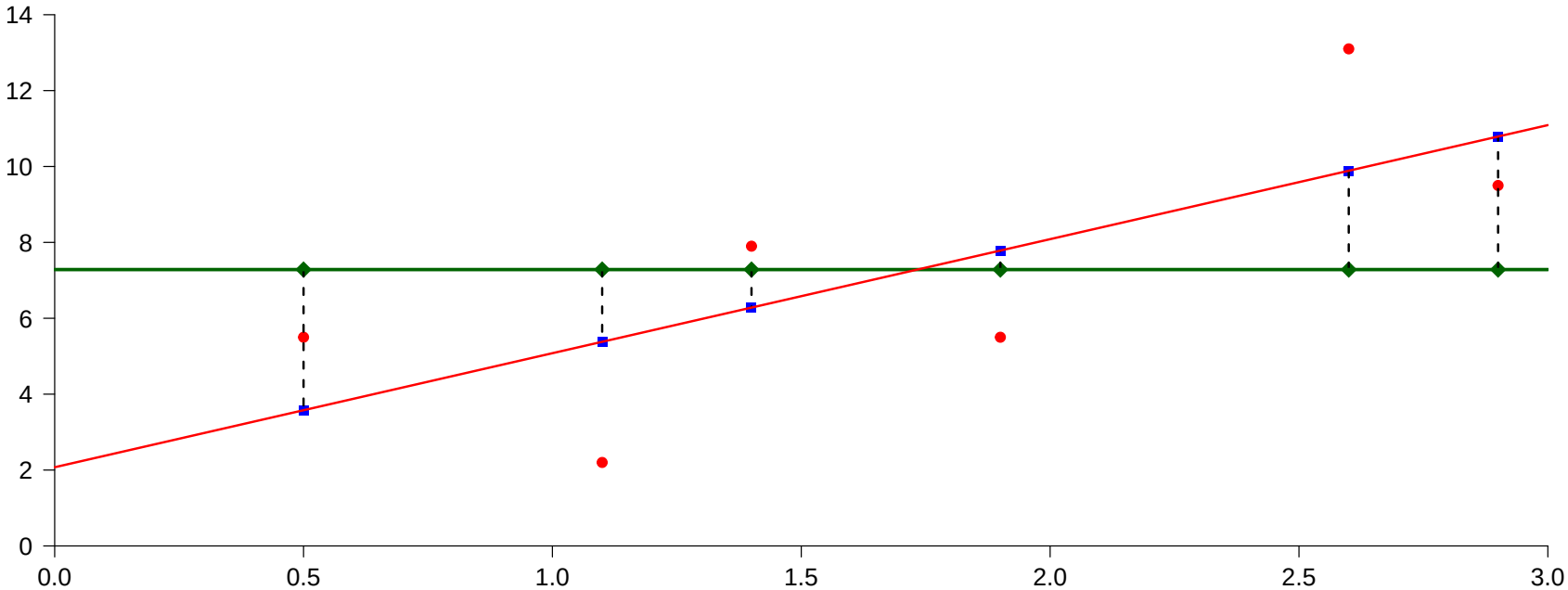
Error sum of squares (ESS)



$$ESS = \sum (y_i - \widehat{y})^2 = \sum \epsilon_i^2$$
$$= \sum \left(y_i - (\beta_0 + \beta_1 x_i) \right)^2$$

1. **red points**: y_i, observations
2. **blue points**: $\widehat{y}_i$, fitted values
3. **dashed lines**: $y_i - \widehat{y}_i$

Regression sum of squares (RSS)



$$RSS = \sum(\widehat{y} - \overline{y})^2$$

1. **red points**: y_i, observations
2. **blue points**: $\widehat{y}_i$, fitted values
3. **green points**: $\overline{y}$, mean value of y
4. **dashed lines**: $\widehat{y} - \overline{y}$

R^2 value

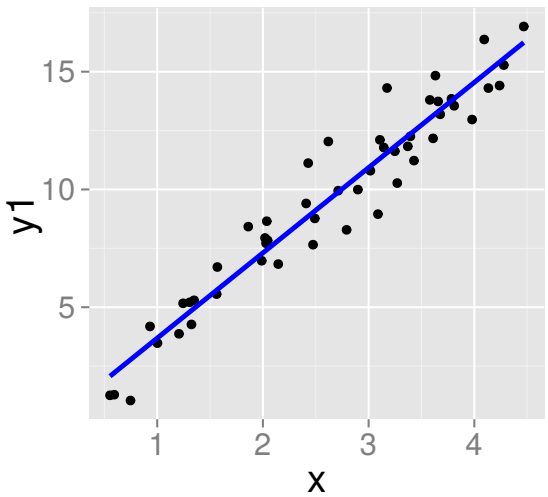
```
> f <- lm (y ~ x)
> summary(f)

Call:
lm(formula = y ~ x)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    2.074      2.730   0.760    0.490
x              3.006      1.419   2.118    0.102

Residual standard error: 2.899 on 4 degrees of freedom
Multiple R-squared:  0.5285, Adjusted R-squared:  0.4107
F-statistic: 4.484 on 1 and 4 DF,  p-value: 0.1016
```

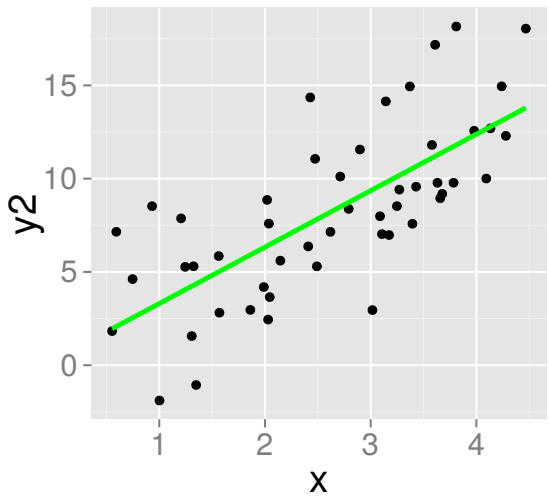
$$R^2 = 0.5285$$

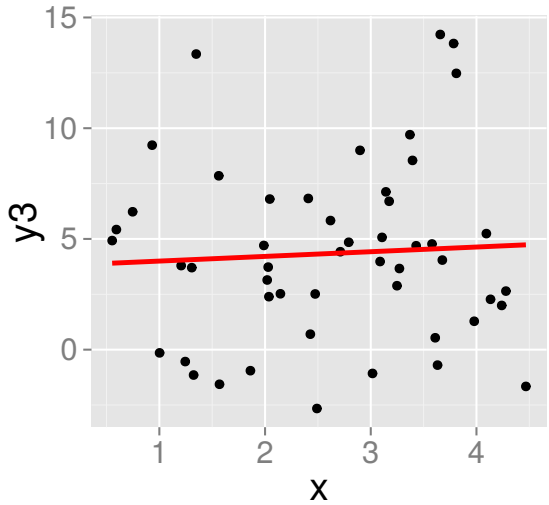
Compute TSS, ESS, RSS and R^2 values

```
> tss <- sum( ( y - mean(y) )^2 )
> rss <- sum( ( f$fit - mean(y) )^2 )
> ess <- sum( ( f$fit - y )^2 )
> r2 <- 1 - ess/tss
> r2
[1] 0.5285435
> r2 <- rss/tss
> r2
[1] 0.5285435
> tss
[1] 71.32833
> rss+ess
[1] 71.32833
```

Comparison with R2








```
lm1 <- lm(y1 ~ x)
lm2 <- lm(y2 ~ x)
lm3 <- lm(y3 ~ x)
r2.1 <- summary(lm1)$r.sq
r2.2 <- summary(lm2)$r.sq
r2.3 <- summary(lm3)$r.sq
> cbind(r2.1, r2.2, r2.3)
         r2.1    r2.2      r2.3
[1,] 0.92725 0.51492 0.0031825
```

R^2 values

Bigger R^2 values **indicate** better fit of the data.

Caution: it is not a strong criterion.

Adjusted R^2

$$\bar{R}^2 = 1 - \frac{n-1}{n-k}\left(1 - R^2\right)$$

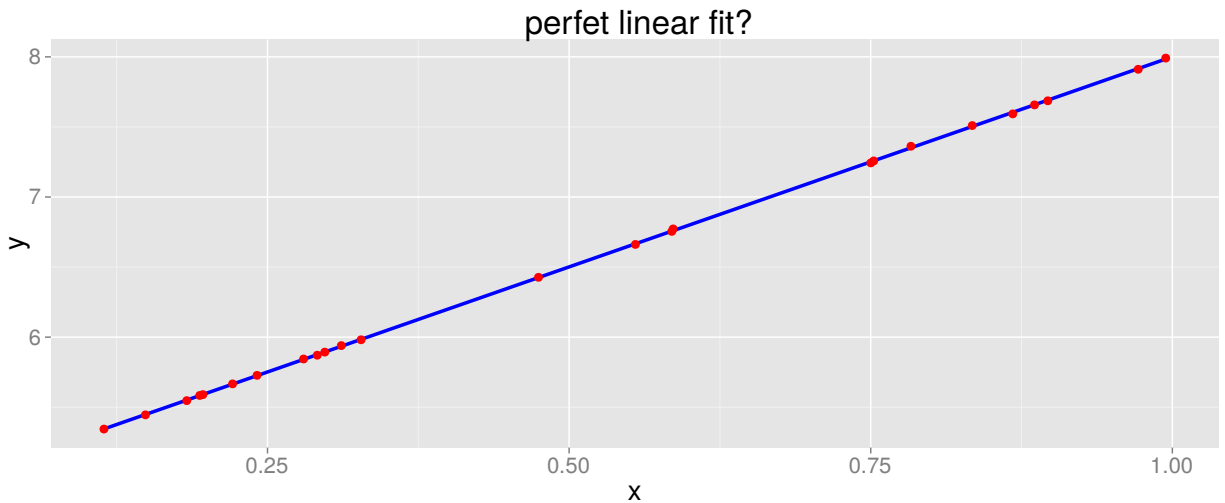
Extract R^2 and $\bar{R}^2$

```
summary(f)$r.sq
summary(f)$adj.r.sq

fs        <- summary(f)
r.sq      <- fs$r.squared
r.sq.adj  <- fs$adj.r.sq
> cbind(r.sq, r.sq.adj)
           r.sq  r.sq.adj
[1,] 0.5285435 0.5893206
```

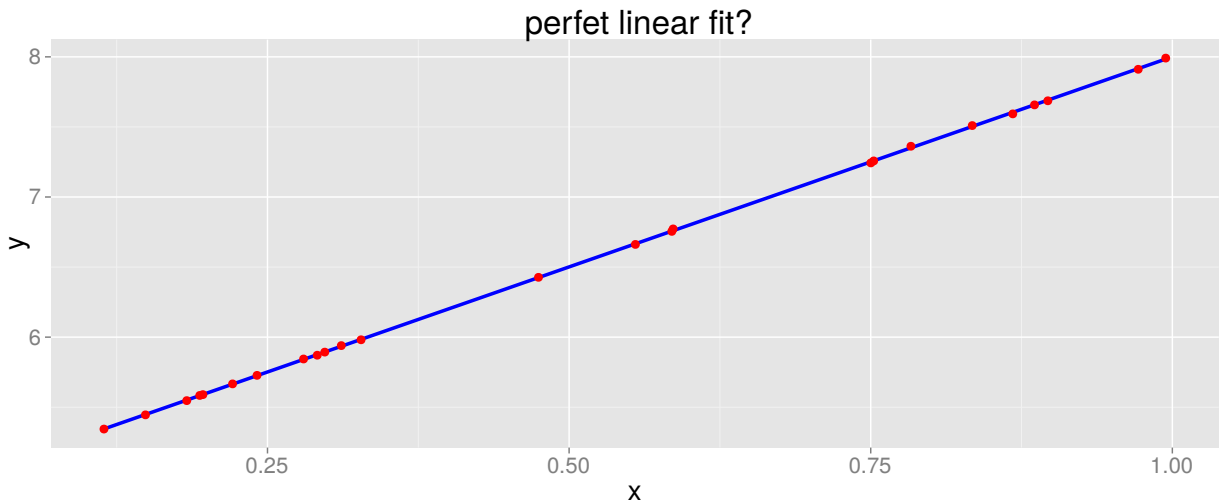
Analytically computing $\bar{R}^2$

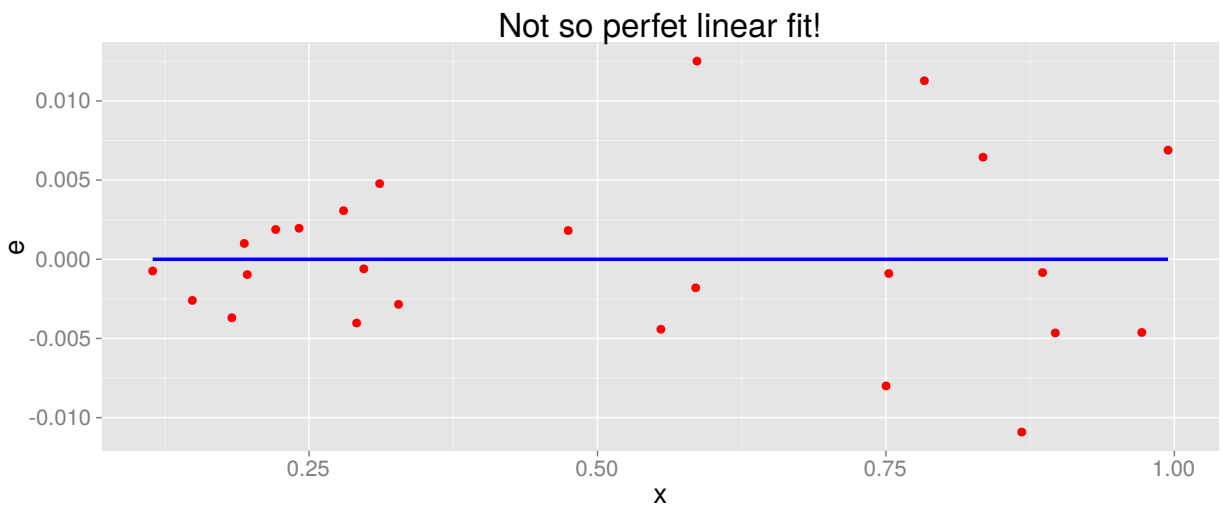
```
n <- length(x)
k <- f$rank
r.sq.adj2 <- (n-1)/(n-k)*(1-r.sq)
> cbind(r.sq.adj, r.sq.adj2)
      r.sq.adj r.sq.adj2
[1,] 0.5893206 0.5893206
```

The heteroscedasticity problem



The heteroscedasticity problem





Comment about correlations in errors

```
> n <- 25
> x <- c(runif(n, 0, 1))
> y <- 5 + 3*x + rnorm(n, sd=x*0.01)
> f <- lm( y~ x )
> summary(f)

Call:
lm(formula = y ~ x)

             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.998397   0.001348    3708   <2e-16 ***
x           3.004048   0.002698    1113   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.003483 on 23 degrees of freedom
Multiple R-squared:      1, Adjusted R-squared:      1
F-statistic: 1.24e+06 on 1 and 23 DF,  p-value: < 2.2e-16
```

Contents

Predicting in `lm` models

```
x  <- c(0.5, 1.1, 1.4, 1.9, 2.6, 2.9)
y  <- c(5.5, 2.2, 7.9, 5.5, 13.1, 9.5)
f  <- lm(y ~ x)
b0 <- f$coef[[1]]
b1 <- f$coef[[2]]
xp <- 2
yp <- b0 + b1*xp
> yp
[1] 8.084824
```

Point prediction

Suppose we have a point $X_p = 2$. We want to know what is the predicted value by the model. This is:

$$Y_p = \widehat{\beta}_0 + \widehat{\beta}_1 \times X_p$$

The above code makes the needed computation.

Prediction in R with multiple values

Using a vector

```
xp <- c(1, 2, 3)
yp <- b0 + b1*xp
yp
[1]  5.079233  8.084824 11.090415
```

Using the predict() function

```
xp <- data.frame(x=c(1, 2, 3))
> predict.lm(f, xp)
        1         2         3
 5.079233  8.084824 11.090415
```

The second option has to be a **data frame**.

Prediction in R with predict

```
> predict(f, interval="prediction")
       fit        lwr      upr
1  3.576438 -6.3849494 13.53782
2  5.379792 -3.6665849 14.42617
3  6.281470 -2.5124817 15.07542
4  7.784265 -0.9357985 16.50433
5  9.888179  0.5462288 19.23013
6 10.789856  0.9539719 20.62574

> newX <- data.frame(x=c(1, 2, 3))
> predict(f, newX)
       1        2        3
 5.079233  8.084824 11.090415

> predict(f, newX, interval="prediction")
       fit        lwr      upr
1  5.079233 -4.0836929 14.24216
2  8.084824 -0.6737388 16.84339
3 11.090415  1.0642844 21.11655
```

Prediction interval

```
fit <- lm (y ~ x)
predict(fit, newdata, interval, level)
```

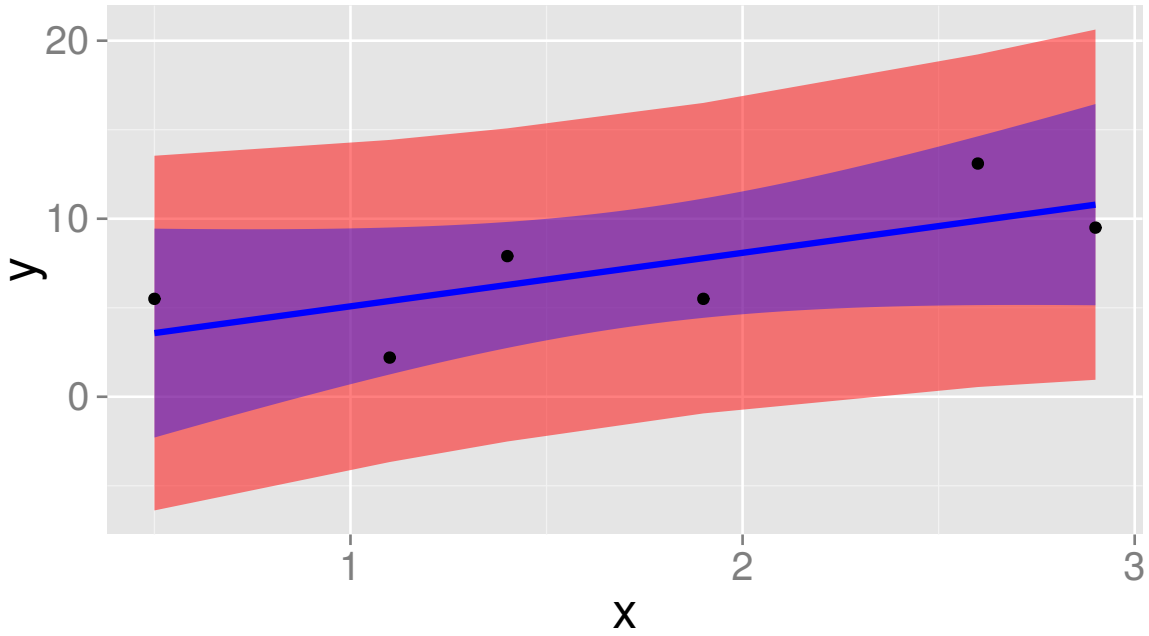
fit	An lm object
newdata	data for prediction (must be in data frame format)
interval	can be "confidence" for confidence interval or "prediction" for prediction interval
level	tolerance/confidence level

Under the assumption of normally distributed residuals the prediction interval can be calculated as:

$$\widehat{Y}_P = \widehat{Y} \pm \widehat{\sigma} \sqrt{1 + \frac{1}{n}\left(1 + \frac{(x_P - \overline{x})^2}{\tilde{S}_x^2}\right)} \cdot t_{n-2;1-\alpha/2}$$

with $\widehat{\sigma} = \frac{1}{n-2} \cdot \sum \widehat{\epsilon}_i^2$ and $\tilde{S}^2 = \frac{n-1}{n} \cdot S^2$

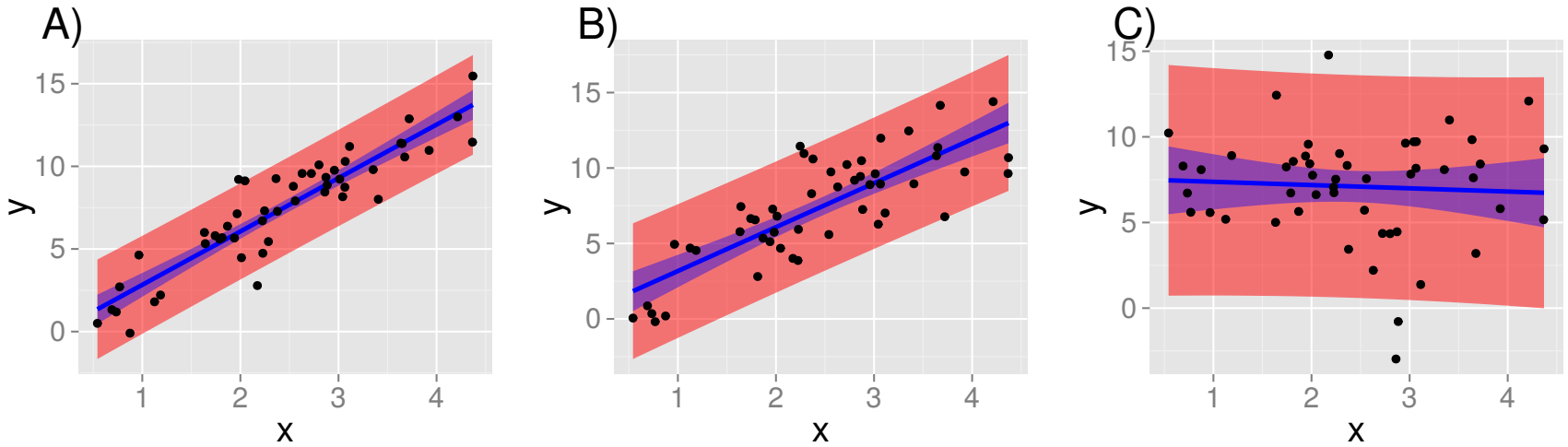
Confidence interval vs Prediction interval



Notes and explanations

1. Confidence interval tells us how good is the fitting.
2. Prediction interval tells where new data points are expected.
3. Confidence : about existed data
   Prediction : about non existed data
4. Prediction interval area is always wider than confidence interval area.

Confidence interval vs Prediction interval



Discussion

Examine the three plots above and comment on the following:

1. Variance of y and ϵ
2. R^2
3. Std. Error of the coefficient (slope, $\widehat{\beta_1}$)
4. P–value of the coefficient (slope, $\widehat{\beta_1}$)
5. Band width of confidence interval
6. Band width of prediction interval

Reading list


J. Verzani g, Using R for Introductory Statistics , 2rd ed., 2014, Chapman & Hall/CRC


J. Fox & S. Weisberg, An R Companion to Applied Regression, 3rd ed., 2016, Sage Publications


C. Kleiber & A. Zeileis, Applied Econometrics with R, 2008, Springer

Online and electronic resources

1. http://www.ats.ucla.edu/stat/r/
2. http://www.stat.wisc.edu/~larget/stat302/
3. https://www.coursera.org/learn/regression-models/

Comments and Questions

Thank you very much

for your attention!