Week 12:

Data Analysis and Wrap-up

07.12.2016

Wednesday 13:00 – 15:00, BM A3

Emre Ugur, BM 33

emre.ugur@boun.edu.tr

cmpe140@listeci.cmpe.boun.edu.tr

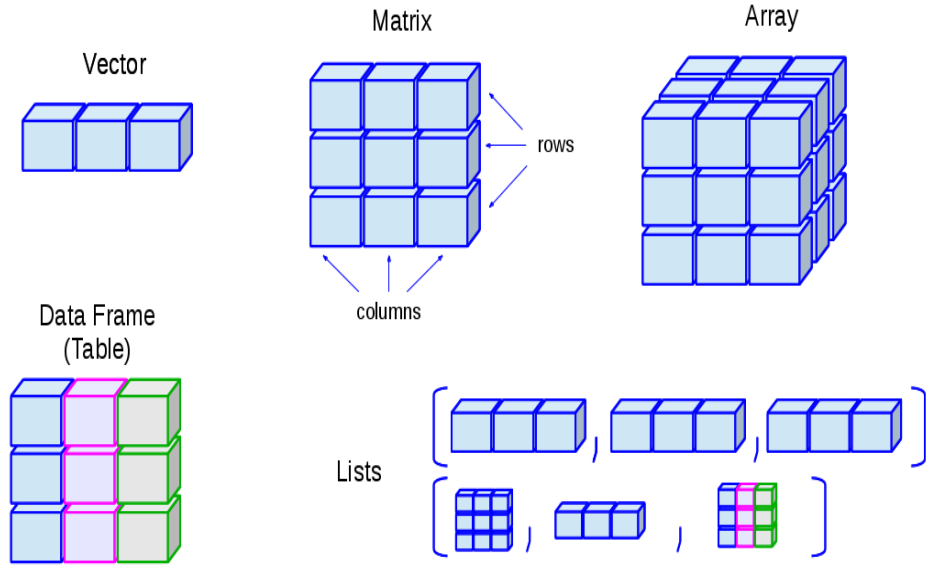
Today

- Data types in R
- Correlation
- Regression
- Wrap-up

- No P.S. today
- There is lab and quiz on Friday

Data structure types

- Vectors: one-dimensional arrays
  - Numeric vectors
  - Complex vectors
  - Logical vectors
  - Character vectors or text strings
- Matrices: two-dimensional
- Arrays: multi-dimensional
- Factors: vectors of categorical variables
- Lists: ordered collection of objects
- Data frames: generalization of matrices, different columns can store different mode data
- Functions: objects that make specific operations



Factors

- Categorical variables
  - Nominal (no order). e.g. `Diabetes(Type1,Type2)`
  - Ordinal (with order). e.g. `Status(poor,improved,excellent)`
- Function `factor()` stores the categorical values as integer internally

```
diabetes <- c("Type1","Type2","Type1","Type1")
diabetes <- factor(diabetes)
```

Internally store as `(1,2,1,1)`.

```
status <- c("Poor","Improved","Excellent","Poor")
status <- factor(status, ordered=TRUE)
```

Internally store as `(3,2,1,3)`.

- Each categorical value corresponds to one **level**.

Factors example 1

```
patientId <- c(1,2,3,4)
age <- c(25,34,28,52)
diabetes <- c("Type1","Type2","Type1","Type1")
diabetes <- factor(diabetes)
status <- c("Poor","Improved","Excellent","Poor")
status <- factor(status,order=TRUE)
patientdata <- data.frame(patientId,age,diabetes,status)
str(patientdata)

'data.frame': 4 obs. of  4 variables:
 $ patientId: num  1 2 3 4
 $ age      : num  25 34 28 52
 $ diabetes : Factor w/ 2 levels "Type1","Type2": 1 2 1 1
 $ status   : Ord.factor w/ 3 levels
"Excellent"<"Improved"<..: 3 2 1 3
```

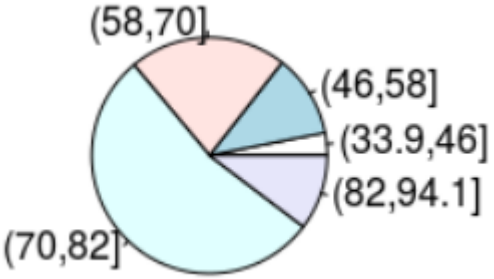
Factors example 1

```
patientId <- c(1,2,3,4)
age <- c(25,34,28,52)
diabetes <- c("Type1","Type2","Type1","Type1")
diabetes <- factor(diabetes)
status <- c("Poor","Improved","Excellent","Poor")
status <- factor(status,order=TRUE)
patientdata <- data.frame(patientId,age,diabetes,status)
summary(patientdata)
```

```
  patientId          age          diabetes       status
 Min.   :1.00   Min.   :25.00   Type1:3   Excellent:1
 1st Qu.:1.75   1st Qu.:27.25   Type2:1   Improved :1
 Median :2.50   Median :31.00             Poor     :2
 Mean   :2.50   Mean   :34.75
 3rd Qu.:3.25   3rd Qu.:38.50
 Max.   :4.00   Max.   :52.00
```

Factors example 2

```
scores <- read.table("scores.txt",header=TRUE)
scores.cut <- cut(scores$Score,breaks=5)
scores.table <- table(scores.cut)
pie (scores.table)
```




```
> str(scores.cut)
 Factor w/ 5 levels "(33.9,46]","(46,58]",..: 5 4 5 4 4 3 4
4 3 4

> summary(scores.cut)
(33.9,46]   (46,58]   (58,70]   (70,82] (82,94.1]
        3        11        21        52        10
```

Conversion between data types

```
as.numeric(), as.double(), as.integer(),
as.logical(), as.character(), as.matrix(),
as.data.frame()  …
```

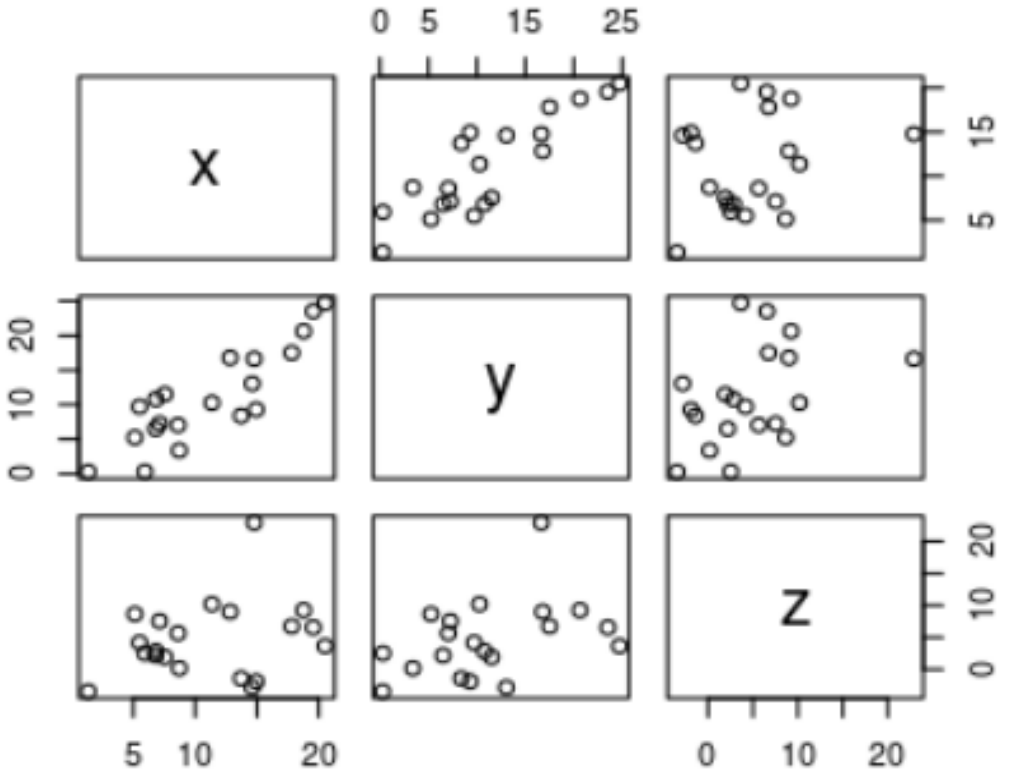
```
as.numeric("kjlsa")
[1] NA
as.numeric("555")
[1] 555
as.logical("555")
[1] NA
as.logical("TRUE")
[1] TRUE
```

Data analysis: statistical relationship

- Correlation looks at global movement shared between two variables
- One variable increases and the other increases as well, then these two variables are said to be positively correlated.
- When a variable increase and the other decrease then these two variables are negatively correlated
- In the case of no correlation no pattern will be seen between the two variable.

Statistical relationship: correlation

```
x<-sample(1:20,20)+rnorm(10,sd=2)
y<-x+rnorm(10,sd=3)
z<-(sample(1:20,20)/2)+rnorm(20,sd=5)
df<-data.frame(x,y,z)
plot(df[,1:3])
```



Statistical relationship: correlation

- Pearson coefficient: the covariance of the two variable divided by the product of their variance
- Scaled between 1 (for a perfect positive correlation) to -1 (for a perfect negative correlation), 0 would be complete randomness.

```
cor(df,method="pearson")
          x           y           z
x  1.0000000  0.8736874 -0.2485967
y  0.8736874  1.0000000 -0.2376243
Z -0.2485967 -0.2376243  1.0000000
```

- From these outputs our suspicion is confirmed x and y have a high positive correlation
- But in statistics we can test if this coefficient is significant.

```
cor.test(df$x,df$y,method="pearson")
```

Modeling statistical relationship: Regression

- Linear regression analysis is the most widely used of all statistical techniques
- Regression analysis is a statistical process for estimating the relationships among variables.
- Focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors')
- Regression analysis is widely used for prediction and forecasting,
- Explain causal relationship between them, for example the most simple linear equation is written : $Y=aX+b$

Linear regression

```
x<-sample(1:20,20)+rnorm(10,sd=2)

y<-x+rnorm(10,sd=3)

z<-(sample(1:20,20)/2)+rnorm(20,sd=5)

df<-data.frame(z,y,x)
```

lm {stats} R Documentation

Fitting Linear Models

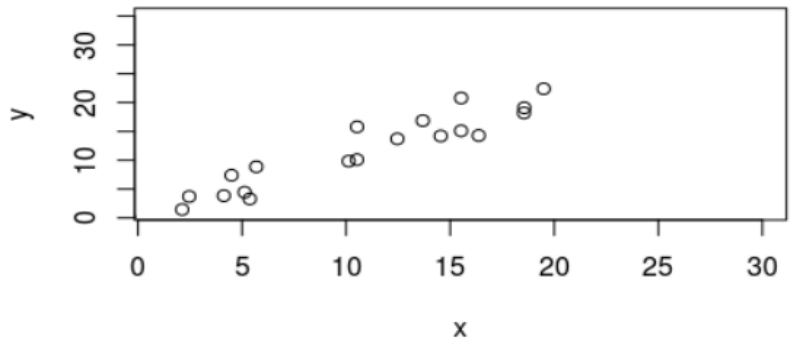
Description

lm is used to fit linear models. It can be used to carry out regression, single stratum analysis of variance and analysis of covariance (although aov may provide a more convenient interface for these).

Usage

```
lm(formula, data, subset, weights, na.action,
   method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE,
   singular.ok = TRUE, contrasts = NULL, offset, ...)
```

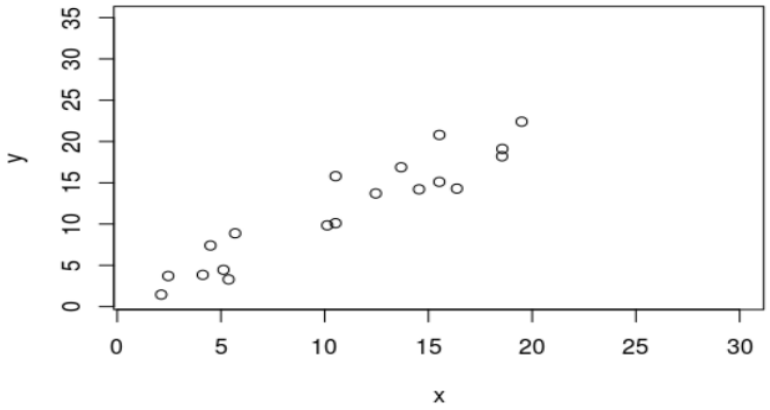
```
plot(x,y,xlim=c(1,30),ylim=c(1,35))
```




```
myModel <- lm(y~x, data=df)
```

Linear regression

```
> plot(x,y,xlim=c(1,30),ylim=c(1,35))
```

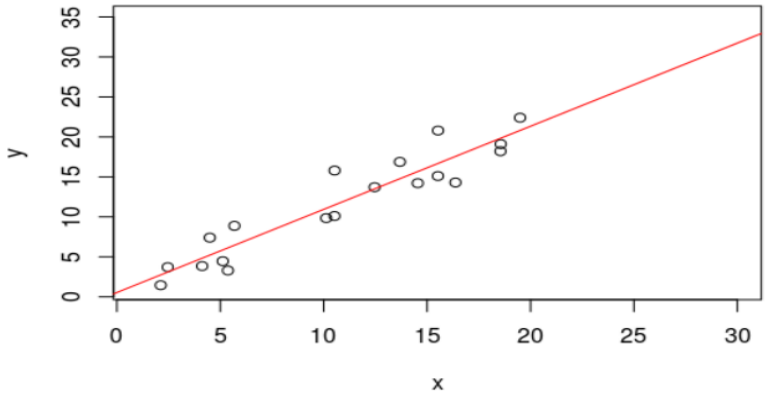



```
> myModel <- lm(y~x, data=df)
```

```
> myModel

Call:

lm(formula = y ~ x, data = trainData)


Coefficients:

(Intercept)                x

     0.5357           1.0397

# model is: y=0.5357+1.0397*x
```

Linear regression

```
> plot(x,y,xlim=c(1,30),ylim=c(1,35))
```

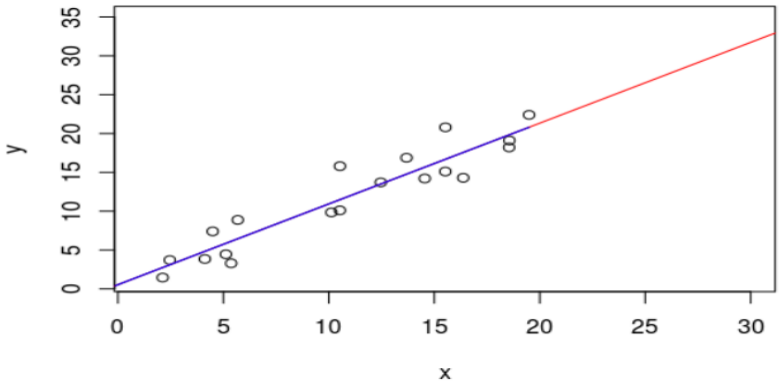



```
> myModel <- lm(y~x, data=df)
```

```
> abline(myModel,col="red")
> str(myModel)
List of 12
 $ coefficients : Named num [1:2] 0.536 1.04
  ..- attr(*, "names")= chr [1:2] "(Intercept)" "x"
 $ residuals    : Named num [1:20] -1.21 4.12 -1.37 -1.45
-2.82 ...
  ..- attr(*, "names")= chr [1:20] "1" "2" "3" "4" ..
# myModel is a list of 12 elements, use the first element
```

Linear regression

```
> plot(x,y,xlim=c(1,30),ylim=c(1,35))
```



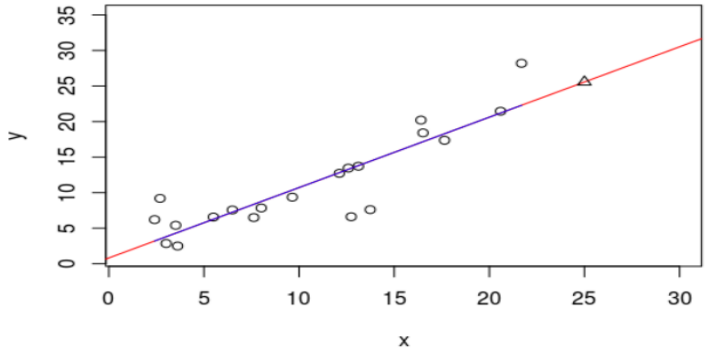

```
> myModel <- lm(y~x, data=df)
```

```
> abline(myModel,col="red")
```

```
> minMaxX <- c(min(x),max(x))
> minY <- myModel[[1]][1] + myModel[[1]][2]*min(x)
> maxY <- myModel[[1]][1] + myModel[[1]][2]*max(x)
> lines(c(min(x),max(x)),c(minY,maxY),col="blue")
```

Linear regression

```
> plot(x,y,xlim=c(1,30),ylim=c(1,35))
> myModel <- lm(y~x, data=df)
> abline(myModel,col="red")
> minMaxX <- c(min(x),max(x))
> y.of.minMaxX <- myModel[[1]][1] + myModel[[1]][2]*minMaxX
> lines(minMaxX,y.of.minMaxX,col="blue")

> newX<-c(25)
> testData <- data.frame(x=newX)
> predY <- predict.lm(myModel,testData)
> points(newX, predY,pch=2)
```



Linear regression

```
> plot(x,y,xlim=c(1,30),ylim=c(1,35))


> myModel <- lm(y~x+z, data=df)


> myModel

Call:

lm(formula = y ~ x + z, data = df)


Coefficients:

(Intercept)                x                z

    0.57456          0.97963          0.06703

# model is: y=0.5357+0.97863*x+0.06703*z
```

Linear regression, real world

```
library(foreign)
library(MASS)
cdata <- read.dta("http://www.ats.ucla.edu/stat/data/crime.dta")
str(cdata)
'data.frame':  51 obs. of  9 variables:
```

- state id (`sid`), state name (`state`), violent crimes per 100,000 people (`crime`), murders per 1,000,000 (`murder`), the percent of the population living in metropolitan areas (`pctmetro`), the percent of the population that is white (`pctwhite`), percent of population with a high school education or above (`pcths`), percent of population living under poverty line (`poverty`), and percent of population that are single parents (`single`).
- It has 51 observations.
- We are going to use poverty and single to predict crime:

```
??????
```

Linear regression, real world

```
library(foreign)
library(MASS)
cdata <- read.dta("http://www.ats.ucla.edu/stat/data/crime.dta")
str(cdata)
'data.frame':  51 obs. of  9 variables:
```

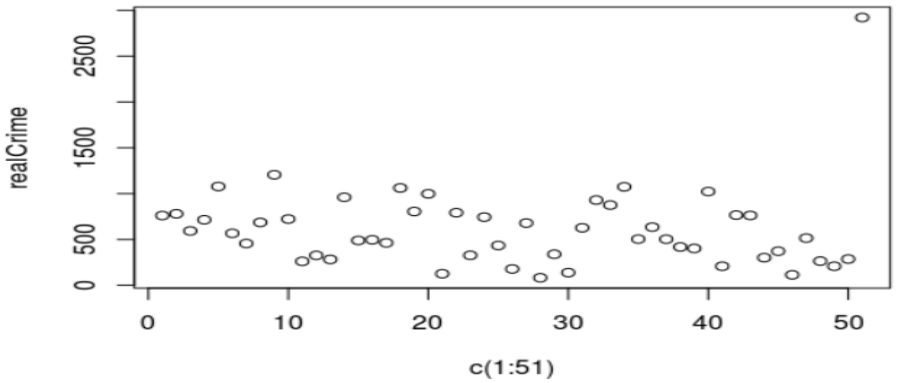
- state id (`sid`), state name (`state`), violent crimes per 100,000 people (`crime`), murders per 1,000,000 (`murder`), the percent of the population living in metropolitan areas (`pctmetro`), the percent of the population that is white (`pctwhite`), percent of population with a high school education or above (`pcths`), percent of population living under poverty line (`poverty`), and percent of population that are single parents (`single`).
- It has 51 observations.
- We are going to use poverty and single to predict crime.

```
crimeModel <- lm(crime ~ poverty + single, data = cdata)
```

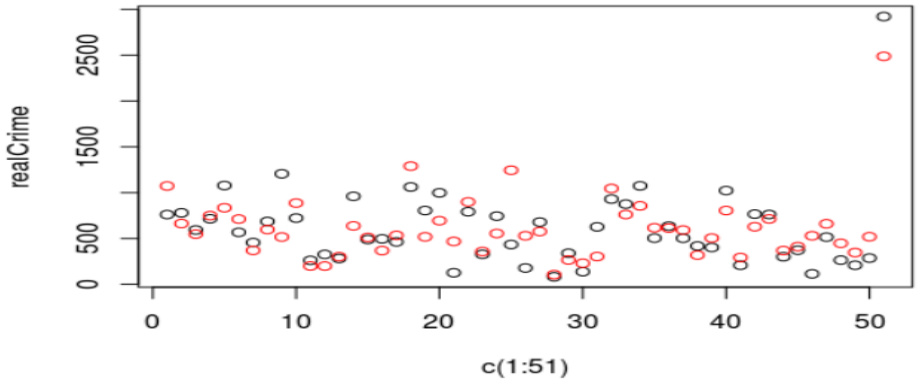
Linear regression, real world

Let us check out how our regression model performs:

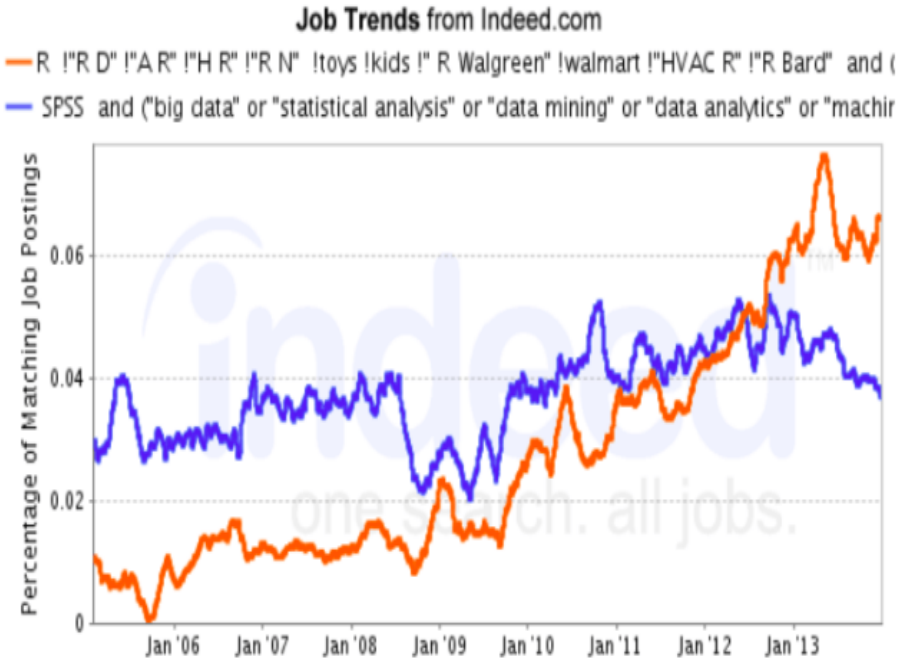
```
realCrime <- cdata$crime
crimeModel <- lm(crime ~ poverty + single, data = cdata)
plot(c(1:51),realCrime)
```




```
testData <- data.frame(poverty=cdata$poverty, single=cdata$single)
predCrime <- predict(crimeModel, testData)
points(c(1:51),predCrime,col=red)
```



Analytics softwares - trends



Analytics job trends for R and SPSS. Note that the legend labels at the top of the graph are truncated due to the very long size of the query.

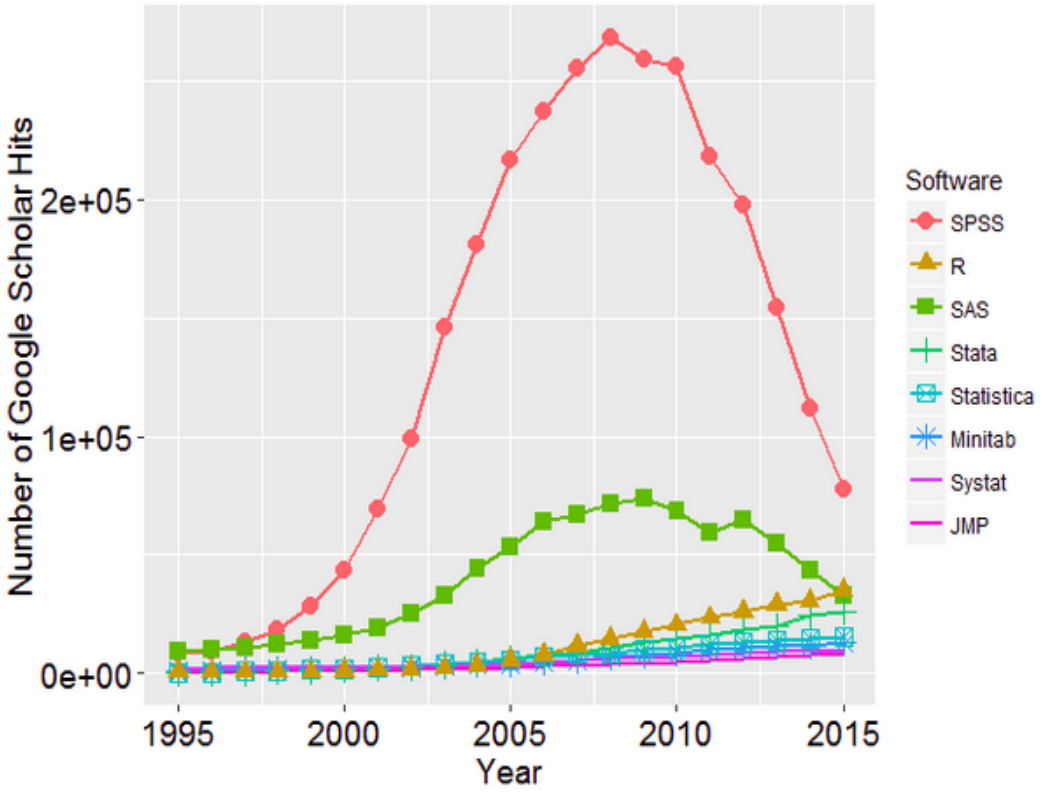


Figure 2d. The number of scholarly articles found in each year by Google Scholar. Only the top six "classic" statistics packages are shown.

Analytical software

Language Rank	Types	Spectrum Ranking
1. C		100.0
2. Java		98.1
3. Python		98.0
4. C++		95.9
5. R		87.9
6. C#		86.7
7. PHP		82.8
8. JavaScript		82.2
9. Ruby		74.5
10. Go		71.9

IEEE Spectrum's Top Ten Languages for 2016

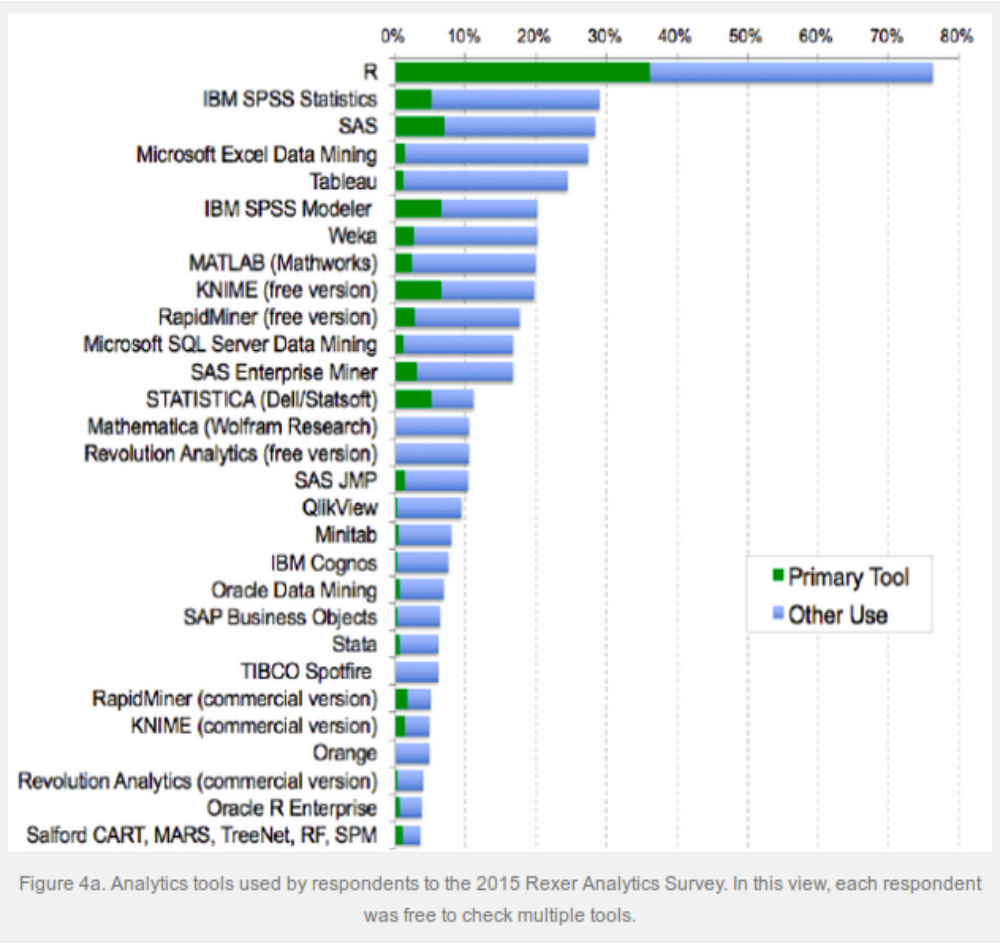


Figure 4a. Analytics tools used by respondents to the 2015 Rexer Analytics Survey. In this view, each respondent was free to check multiple tools.

Surveys of data scientiests.

Next week

- There will be lecture, P.S. and lab
- We will review the previous lectures
- There will be no quiz next week

Homework

- Download grades as .csv file
- Read them into a data frame
- Find the correlation among attandence, midterm and quiz grades
- Repeat crime analysis, use other variables (e.g. `pcths`) for this. Repeat the prediction plot.
  - See how the mean residual (error) changes by
    - `summary(myModel)`
    - `str(myModel)`