Overview of Regression and Correlation

Daniel Lawson University of Bristol

Lecture 02.1.1 (v1.0.1)

Signposting

- Last time we looked at **Exploratory Data Analysis**.
- Correlation is a description of such data, whilst regression is the first tool to reach for when trying to make sense of such an analysis.
- Regression is a **linear** method and as such, it is usually best considered a form of EDA.
- This section is about interpreting Regression[1].
- The following section 02.1.2 is the mathematical content for Modern Regression (Matrix representation).

[1] This is mostly background knowedge. Read up if you are unfamiliar or rusty.

Intended Learning Outcomes

- ILOs used:
  - ILO1 Be able to **access and process cyber security data** into a format suitable for mathematical reasoning
  - ILO2 Be able to **use and apply basic machine learning** tools
  - ILO3 Be able to **make and report appropriate inferences** from the results of applying basic tools to data

Correlation

- **Correlation** is the basic relationship between x and y
- The **linear relationship** in 2-d data.
- Asks, "How does **variation** in x associate with variation in y?"
  - Correlation examines this relationship in a symmetric manner.
  - Consequently, correlation does not attempt to establish any cause and effect.
  - It is a **descriptive statistic**, like variance

Covariance

- A reminder: covariance is simply the second (central) moment:

$$\mathrm{cov}(X, Y) = \mathbb{E}\left[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])\right]$$

- it is straightforward to show that

$$\mathrm{cov}(X, Y) = \mathbb{E}\left[XY\right] - \mathbb{E}\left[X\right]\mathbb{E}\left[Y\right].$$

- Recall that we typically use **unbiased** estimators which often slightly different from natural theoretical analogue. The **sample covariance** is:

$$\mathrm{cov}(X, Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})$$

Correlation

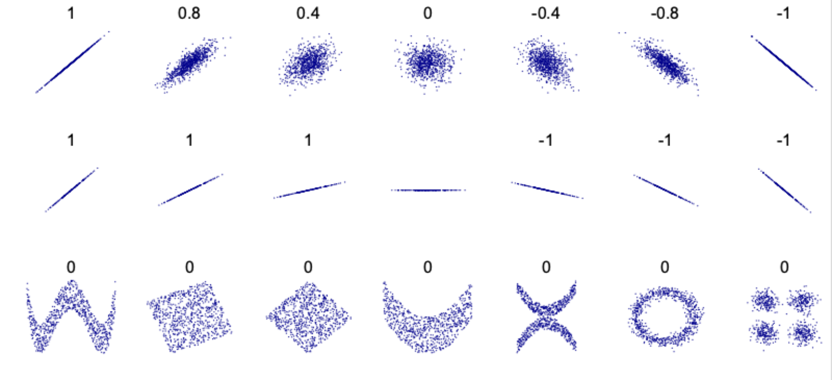
- Correlation is simply a **normalised measure** of covariance.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y}$$

- It takes values between -1 and 1.
- Sample correlation uses the unbiased estimator of covariance, to account for the number of degrees of freedom in the data.
- Advanced topics in correlation include rank correlation, canonical correlation, estimation from correlation matrices, etc.

Examples

From Wikipedia: Correlation_and_dependence



Regression

- **Regression**, considers the relationship of a response variable as determined by one or more explanatory variables.
  - Regression is designed to help **make predictions** of y when we observe x.
  - It is **not** a joint model of x and y.
  - It predicts the *best guess*.
  - There is a probabilistic interpretation based on Normal Distributions.
- Regression is a often used as a tool to establish causality…
  - A and B share a causal relationship if a regression for B given A, conditional on C (C=**everything else**), has an association
  - This does not resolve whether A causes B, or B causes A
  - Since we don't measure **everything else**, regression rarely establishes causality!
  - Assumptions are needed to make a causal connection. This is known as **causal inference** and there are frameworks to establish causality.

Example of correlation

- R code:

```
conncor1=c(linear=cor(conndata2[,'orig_pkts'],
                      conndata2[,'orig_ip_bytes']),
    log=cor(log(1+conndata2[,'orig_pkts']),
                log(1+conndata2[,'orig_ip_bytes'])))
kable(conncor1)
```

Example of correlation

- R code:

```
conncor1=c(linear=cor(conndata2[,'orig_pkts'],
                      conndata2[,'orig_ip_bytes']),
    log=cor(log(1+conndata2[,'orig_pkts']),
                 log(1+conndata2[,'orig_ip_bytes'])))
kable(conncor1)
```

- Which gives:

	Correlation
linear	0.9911887
log	0.9452585

Example of correlation

- R code:

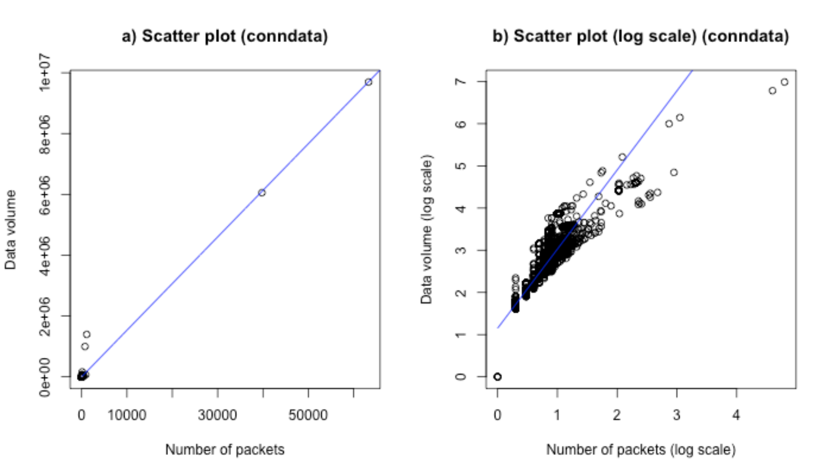
```
conncor1=c(linear=cor(conndata2[,'orig_pkts'],
                      conndata2[,'orig_ip_bytes']),
    log=cor(log(1+conndata2[,'orig_pkts']),
                 log(1+conndata2[,'orig_ip_bytes'])))
kable(conncor1)
```

- Which gives:

	Correlation
linear	0.9911887
log	0.9452585

- Linear-scale correlation is dominated by the large values, which makes it look better than it really is.

EDA for regression



Example of data frame correlation

```r
## Extracting valid data
conndatasize=conndata2[,c('orig_pkts','resp_pkts',
                    'orig_bytes','resp_bytes')]
## Removing missing data
conndatasize=conndatasize[
    !apply(conndatasize,1,function(x)any(x=="-")),]
## Converting to numeric
for(i in 1:dim(conndatasize)[2])
conndatasize[,i]=as.integer(conndatasize[,i])
```

Example of data frame correlation

```
## Extracting valid data
conndatasize=conndata2[,c('orig_pkts','resp_pkts',
                       'orig_bytes','resp_bytes')]
## Removing missing data
conndatasize=conndatasize[
    !apply(conndatasize,1,function(x)any(x=="-")),]
## Converting to numeric
for(i in 1:dim(conndatasize)[2])
conndatasize[,i]=as.integer(conndatasize[,i])

## Computing the correlation matrix
cordatasize=cor(conndatasize)
```

Example of data frame correlation

	orig_pkts	resp_pkts	orig_bytes	resp_bytes
orig_pkts	1	0.999705713975153	0.00108611529297986	0.00264433365396342
resp_pkts	0.999705713975153	1	0.000945267947070292	0.00262111604209595
orig_bytes	0.00108611529297986	0.000945267947070292	1	0.0735429914375197
resp_bytes	0.00264433365396342	0.00262111604209595	0.0735429914375197	1

R Code for previous slide

```r
## Extracting valid data
library(DT)
library(RColorBrewer)
cuts=seq(0,1,length.out=101)[-1]
colors=colorRampPalette(brewer.pal(9,'Blues'))(101)
datatable(cordatasize) %>%
  formatStyle(columns = rownames(cordatasize),
  background = styleInterval(cuts[1:60],colors[1:61]))
```

Discrete predictors

- If you include categorical/factor predictors, each **level** or unique value is used as a binary predictor.
- Nothing clever is done by default!

Important measures of regression

- **R squared** (and adjusted R squared): variance explained/total variance. This tells us how predictable y is.
- The coefficients β_i, compared to their error $\hat{\sigma}_i$. This is converted to a t-value ($t_i = \beta_i / \hat{\sigma}_i$) and a p-value.
- F statistic and F test p-value: The ratio of the explained to unexplained variance, accounting for the **degrees of freedom**. Your model is compared to a null in which there are no explanatory variables. It is used in variable selection, ANOVA, etc.

A new dataset for average packet size

```
conndatasize2=conndata2[,c('orig_pkts','resp_pkts','orig_bytes',
     'resp_bytes','service')]
## Missing data
conndatasize2=conndatasize2[!apply(conndatasize2,1,
                                   function(x)any(x=="-")),]
## Average packet size
for(i in 1:4)
  conndatasize2[,i]=as.integer(conndatasize2[,i])
  conndatasize2$orig_avg_size=
    conndatasize2$orig_bytes/conndatasize2$orig_pkts
  conndatasize2$resp_avg_size=
    conndatasize2$resp_bytes/conndatasize2$resp_pkts
# Make a factor
conndatasize2[,'service']=as.factor(conndatasize2[,'service'])
for(i in 1:4) # log-transform raw data
    conndatasize2[,i]=log(1+conndatasize2[,i])
```

Linear Modelling in R: average packet size

```
lm(orig_avg_size~resp_avg_size+orig_pkts+
        orig_bytes,data=conndatasize2) %>% summary

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)   -1.663e+02  4.156e+00 -40.025   <2e-16 ***
resp_avg_size -1.039e-07  2.388e-07  -0.435    0.664
orig_pkts     -4.172e+01  1.743e+00 -23.928   <2e-16 ***
orig_bytes     5.330e+01  1.067e+00  49.960   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 43.77 on 4462 degrees of freedom
Multiple R-squared:  0.4231,	Adjusted R-squared:  0.4227
F-statistic:  1091 on 3 and 4462 DF,  p-value: < 2.2e-16
```

Linear Modelling in R: average packet size

```
summary(lm(orig_avg_size~resp_avg_size+orig_pkts+
              orig_bytes+service,data=conndatasize2))

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -1.119e+02  2.371e+01  -4.719 2.44e-06 ***
resp_avg_size   -4.301e-08  2.233e-07  -0.193   0.8472
orig_pkts       -6.259e+01  1.843e+00 -33.956  < 2e-16 ***
orig_bytes       6.985e+01  1.193e+00  58.544  < 2e-16 ***
serviceftp       1.059e+01  2.504e+01   0.423   0.6723
serviceftp-data  2.120e+02  2.748e+01   7.715 1.49e-14 ***
servicehttp     -1.111e+02  2.379e+01  -4.669 3.12e-06 ***
servicesmtp      8.929e+01  3.735e+01   2.390   0.0169 *
servicessh      -5.968e+01  2.438e+01  -2.448   0.0144 *
servicessl      -1.189e+02  2.387e+01  -4.982 6.52e-07 ***
```

Comparing models

```
lm(resp_avg_size~.,data=conndatasize2) %>% summary

Multiple R-squared:  0.05945,   Adjusted R-squared:  0.05712
F-statistic: 25.59 on 11 and 4454 DF,  p-value: < 2.2e-16

lm(orig_avg_size~.,data=conndatasize2) %>% summary

Multiple R-squared:  0.5326,    Adjusted R-squared:  0.5315
F-statistic: 461.4 on 11 and 4454 DF,  p-value: < 2.2e-16
```

Linear Modelling in R: average packet size

- Conclusions:
  - Response size is harder to predict than Original message size
- For sent packets:
  - Packet size is larger if you send fewer packets, or more data
  - HTTP, SSH and SSL all send smaller packets than DNS, SMTP, *and FTP*
- Important caveats:
  - **this is all excluding any record containing missing data**
  - Not careful transformations
  - Only true on average

Reflection

- Make sure you really understand univariate regression
  - Be familiar with what it does and does not show
  - Know its practical use and abuse
- The technical content is all pre-requisite material

Signposting

- Further reading:
  - Cosma Shalizi's Modern Regression Lectures (Lectures 4-9)
  - This background is well served by Wikipedia's Linear Regression and numerous textbooks and resources.
- Make sure to look at **02.1-Regression.R**
- Regression is the "basic class" of model. We have something to look into with:
  - **Statistical Testing**,
  - **Resampling** methods, and
  - **Model Selection**
- Next up: the mathematics of Modern Regression