Bush 631-603: Quantitative Methods

Lecture 6 (02.21.2023): Prediction vol. I

Rotem Dvir

The Bush school of Government and Public Policy

Texas A&M University

Spring 2023

What is today's plan?

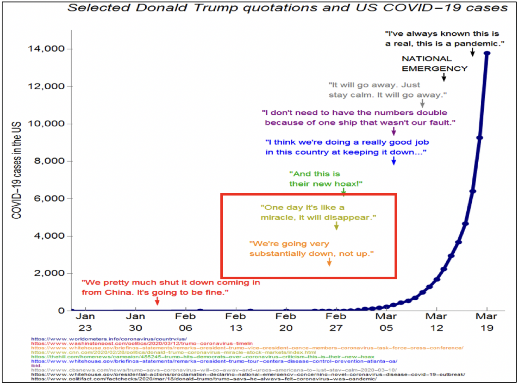
- Why predictions?
- Programming basics - loops, conditional statements.
- Making predictions with data: elections, FP expenses.
- Using dates data.
- R Tech: mastering R Markdown
- R work: loops, if{}, if{}else{}, as.date(), line plots.
- Programming Task: Working with R

Predicting with data

- Social science research:
  - Establish causality.
  - The role of measurement.
- Predictions:
  - Support for causal statements.
  - Generate accurate predictions about potential outcomes.

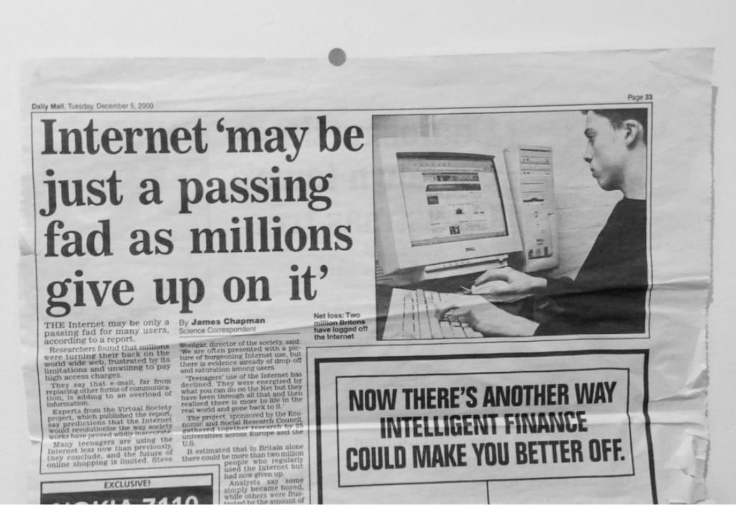
Not the best...predictions!

Oh no...



Some more gems

Daily Mail - December 5, 2000



Daily Mail, Tuesday, December 5, 2000 Page 33

Internet 'may be just a passing fad as millions give up on it'

By **James Chapman**
Science Correspondent

THE Internet may be only a passing fad for many users, according to a report.

Researchers found that millions were turning their back on the world wide web, frustrated by its limitations and unwilling to pay high access charges.

They say that e-mail, far from replacing other forms of communication, is adding to an overload of information.

Experts from the Virtual Society project, which published the report, say predictions that the Internet would revolutionise the way society works have proved wildly inaccurate.

Many teenagers are using the Internet less now than previously, they conclude, and the future of online shopping is limited. Steve Woolgar, director of the society, said: 'We are often presented with a picture of burgeoning Internet use, but there is evidence already of drop-off and saturation among users.

'Teenagers' use of the Internet has declined. They were energised by what you can do on the Net but they have been through all that and then realised there is more to life in the real world and gone back to it.'

The project, sponsored by the Economic and Social Research Council, gathered together research by 25 universities across Europe and the U.S.

It estimated that in Britain alone there could be more than two million people who regularly used the Internet but had now given up.

Analysts say some simply became bored, while others were frustrated by the amount of

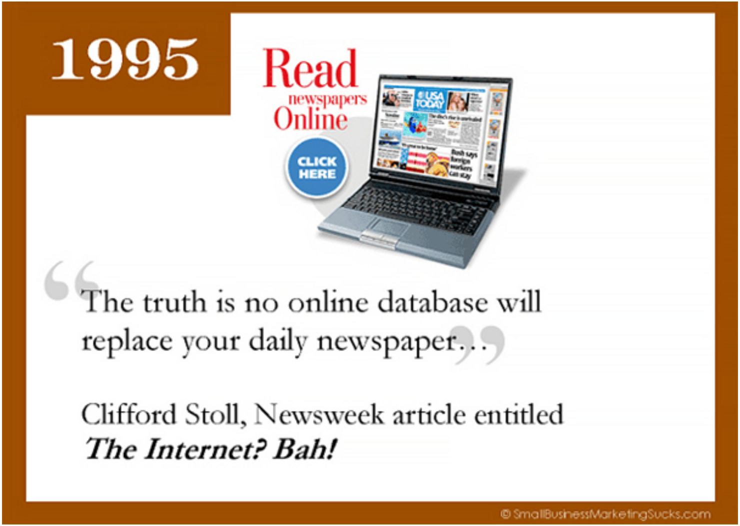
Net loss: Two million Britons have logged off the Internet

EXCLUSIVE!

NOW THERE'S ANOTHER WAY INTELLIGENT FINANCE COULD MAKE YOU BETTER OFF.

Some more gems

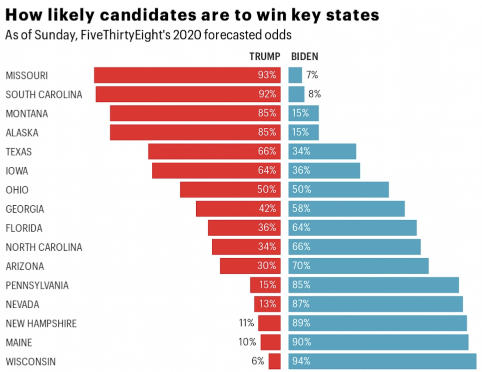
Well. . .



Some groundwork

Loops

- Useful to repeat the same operation multiple times.
- Efficient analysis tool.



Loops in R

- Run similar code chunk repeatedly.

```
for (i in X) {
  expression1
  expression2
  ...
  expression3
}
```

- Elements of loop:
  - i: counter (change as you like).
  - X: Vector of ordered values for the counter.
  - expression: set of expressions to run repeatedly.
  - {}: curly braces define the beginning and end of a loop.

Loops in R

```r
weeks <- c(1,2,3,4,5)
n <- length(weeks)
t <- rep(NA,n)

# loop counter
for (i in 1:n){
  t[i] <- weeks[i] * 2
  cat("I completed Swirl HW number", weeks[i], "in",
      t[i], "minutes", "\n")
}
```

```
## I completed Swirl HW number 1 in 2 minutes
## I completed Swirl HW number 2 in 4 minutes
## I completed Swirl HW number 3 in 6 minutes
## I completed Swirl HW number 4 in 8 minutes
## I completed Swirl HW number 5 in 10 minutes
```

Conditional statements

- Implement code chunks based on logical expressions.

If statements

Syntax: if(x = a condition){set of commands}

Run command(s) only if value if X is TRUE

```
weather <- "rain"
if (weather == "rain"){
  cat("I should take my umbrella")
}
```

```
## I should take my umbrella
```

Flexible if statements

If Else statements

Using if(){} else {}

```
weather <- "sunny"
if (weather == "rain"){
  cat("I should take my umbrella")
} else {
  cat("I should wear my Aggie hat")
}
```

```
## I should wear my Aggie hat
```

Complex conditional statements

Join conditional statements into a loop.

```
days <- 1:7
n <- length(days)

for (i in 1:n){
  x <- days[i]
  r <- x %% 2

  if (r == 0){
    cat("Day", x, "is even and I need my umbrella \n")
  } else {
    cat("Day", x, "is odd and I need my Aggie cap \n")
  }
}
```

```
## Day 1 is odd and I need my Aggie cap
## Day 2 is even and I need my umbrella
## Day 3 is odd and I need my Aggie cap
## Day 4 is even and I need my umbrella
## Day 5 is odd and I need my Aggie cap
## Day 6 is even and I need my umbrella
## Day 7 is odd and I need my Aggie cap
```

Conditional statements

Integrate conditional statements within a conditional statement.

```r
  output$tab <- function(){

## Season 2016: Tables
    if(input$year == 2016){
      data2016 <- mydata %>%
        filter(season == 2016)

    if (input$data == "QBR") {
      dat_tab <- data2016 %>%
        filter(QBR_rank < 16) %>%
        select(First, Last, QBR)

      dat_tab %>%
        knitr::kable("html") %>%
        kable_styling(font_size = 15, "striped", full_width = F, position = "center") %>%
        add_header_above(c("QBR: Top 15" = 3)) %>%
        scroll_box(height = "250px", width = "450px")
    } else
     if (input$data == "EPA") {
        dat_tab <- data2016 %>%
          filter(EPA_rank < 16) %>%
          select(First, Last, EPA_play) %>%
          arrange(-EPA_play)
```

Conditional statements

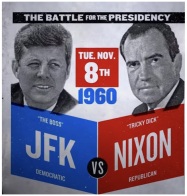
Caution:

- if(){} else{} are complex.
- Double check the curly braces for each statement.
- Use the automatic indentation.
- 'Space-out' your code.
- Add comments (using #) to clearly mark each step.

Predictions

- Awesome research tool. . . with the right design.
- Predict: elections, economic trends, behavior, Superbowl winners, etc.

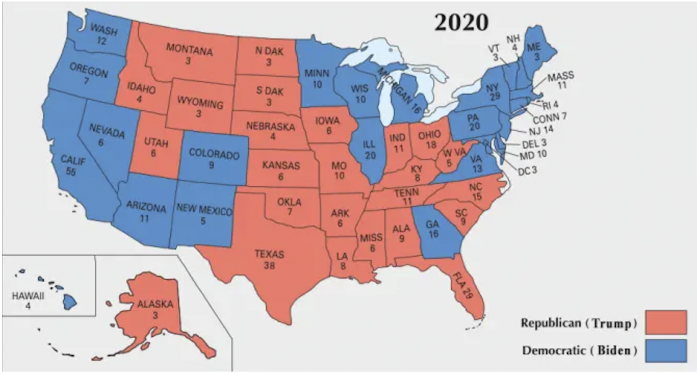
Elections winner



US electoral system

Electoral college

Plurality of votes in a state: “Winner-take-all”



Election predictions

Measurement problem:

- National vote vs. electoral votes.
- Bush - Gore (2000).
- Clinton - Trump (2016).

Electoral vote:

- Number of electors does not align with number of voters per state.
- Votes are “unaccounted”.

A Prediction problem:

- Accurate forecast of **each state** winner.

Polls and election predictions

Data: 2016 elections (polls)

```
head(polls16)
```

```
##   state  middate daysleft                  pollster
## 1    AK  8/11/16       89  Lake Research Partners
## 2    AK  8/20/16       80              SurveyMonkey
## 3    AK 10/20/16       19                    YouGov
## 4    AK 10/26/16       13 Google Consumer Surveys
## 5    AK  9/30/16       39 Google Consumer Surveys
## 6    AK 10/12/16       27 Google Consumer Surveys
##   clinton trump margin
## 1    30.0  38.0   8.00
## 2    31.0  38.0   7.00
## 3    37.4  37.7   0.30
## 4    38.0  39.0   1.00
## 5    47.5  36.7 -10.76
## 6    34.6  30.0  -4.62
```

Poll prediction by states (using R loop)

```r
poll.pred <- rep(NA, 51) # place holder

# get list of unique state names to iterate over
st.names <- unique(polls16$state)

# add labels to holder
names(poll.pred) <- st.names

for (i in 1:51) {
  state.data <- subset(polls16, subset = (state == st.names[i]))

  latest <- state.data$daysleft == min(state.data$daysleft)

  poll.pred[i] <- mean(state.data$margin[latest])
}

head(poll.pred)
```

```
##     AK     AL     AR     AZ     CA     CO
##  14.73  29.72  20.02   2.50 -23.00  -7.05
```

Errors in polling

Prediction error = actual outcome - predicted outcome

```
errors <- pres16$margin - poll.pred
names(errors) <- st.names
mean(errors)
```

```
## [1] 3.81
```

Root mean-square-error (RMSE): average magnitude of prediction error

```
sqrt(mean(errors^2))
```

```
## [1] 9.6
```

Prediction challenges

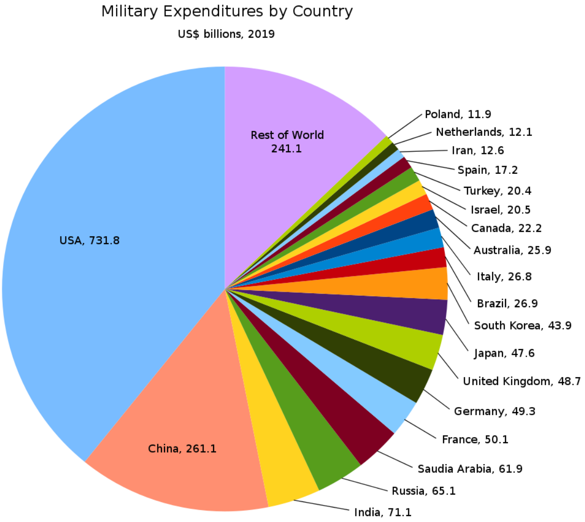
Prediction of binary outcome variable $\rightarrow$ classification problem

Wrong prediction $\rightarrow$ misclassification:

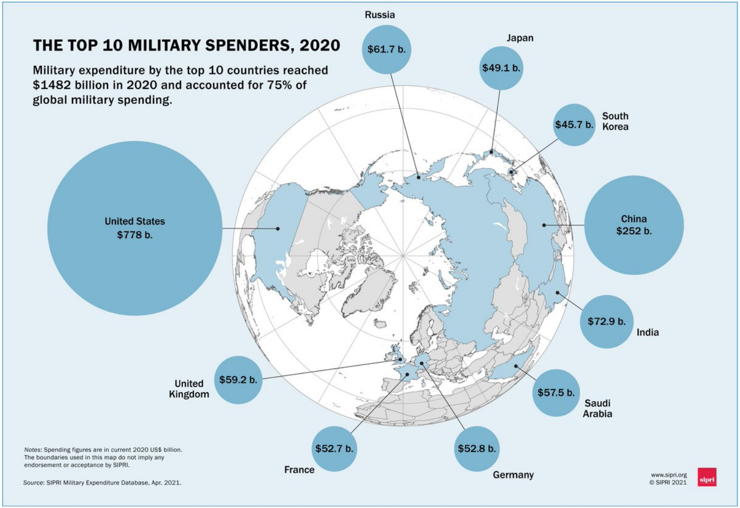
1. true positive: predict Trump wins when he actually wins.
2. **false positive**: predict Trump wins when he actually loses.
3. true negative: predict Trump loses when he actually loses.
4. **false negative**: predict Trump loses when he actually wins.

2016 elections: misclassification rate was high: 9.8% (5/51 states).

Military spending across the globe



Military spending across the globe



Predicting military spending

Our data:

- 157 Countries
- Time frame: 1999-2019
- Measure: military spending as proportion of total gov't spending.

Why this measure?

- Reflect state's preferences.
- Trade-off: *Guns vs. Butter*.

Our predictions:

- Using 1999-2019 data to predict 2020 levels.
- Test predictions with actual data.

Military spending data

```
dim(mil_exp)
```

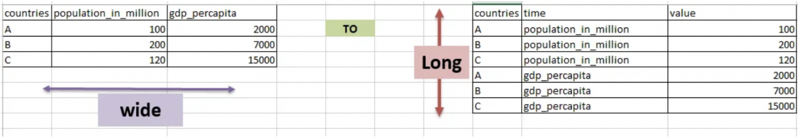
```
## [1] 157  25
```

```
head(mil_exp, n=8)
```

```
## # A tibble: 8 x 25
##   Country Group1 Subgr~1 `1999` `2000` `2001` `2002` `2003` `2004` `2005` `2
##   <chr>   <chr>  <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <
## 1 Algeria Africa North ~ 0.118  0.120  0.122  0.108  0.101  0.107  0.105  0.
## 2 Libya   Africa North ~ 0.115  0.103  0.0630 0.0524 0.0484 0.0490 0.0502 0.
## 3 Morocco Africa North ~ 0.145  0.0898 0.145  0.125  0.134  0.123  0.105  0.
## 4 Tunisia Africa North ~ 0.0618 0.0614 0.0605 0.0590 0.0603 0.0591 0.0601 0.
## 5 Angola  Africa Sub-Sa~ 0.274  0.129  0.108  0.0919 0.109  0.116  0.139  0.
## 6 Benin   Africa Sub-Sa~ 0.0452 0.0264 0.0232 0.0407 0.0473 0.0506 0.0482 0.
## 7 Botswa~ Africa Sub-Sa~ 0.0759 0.0817 0.0899 0.0900 0.0915 0.0848 0.0823 0.
## 8 Burkin~ Africa Sub-Sa~ 0.0576 0.0624 0.0588 0.0605 0.0610 0.0596 0.0594 0.
## # ... with 14 more variables: `2007` <dbl>, `2008` <dbl>, `2009` <dbl>,
## #   `2010` <dbl>, `2011` <dbl>, `2012` <dbl>, `2013` <dbl>, `2014` <dbl>,
## #   `2015` <dbl>, `2016` <dbl>, `2017` <dbl>, `2018` <dbl>, `2019` <dbl>,
## #   `2020` <dbl>, and abbreviated variable name 1: Subgroup1
```

Reshaping the data

- Use the gather() function
- Increase the data size.
- Cross section $\rightarrow$ Panel (TSCS Data).
- Each case (country for us) has multiple observations (rows).



countries	population_in_million	gdp_percapita
A	100	2000
B	200	7000
C	120	15000

countries	time	value
A	population_in_million	100
B	population_in_million	200
C	population_in_million	120
A	gdp_percapita	2000
B	gdp_percapita	7000
C	gdp_percapita	15000

Reshaping the data

gather() function: long-form data.

```
spend_long <- mil_exp2 %>%
  gather(year, exp, '1999':'2019',-Country, -Group1, -Subgroup1) %>%
  arrange(Country)

head(spend_long, n=9)
```

```
## # A tibble: 9 x 5
##   Country     Group1         Subgroup1  year    exp
##   <chr>       <chr>          <chr>      <chr> <dbl>
## 1 Afghanistan Asia & Oceania South Asia 1999  NA
## 2 Afghanistan Asia & Oceania South Asia 2000  NA
## 3 Afghanistan Asia & Oceania South Asia 2001  NA
## 4 Afghanistan Asia & Oceania South Asia 2002  NA
## 5 Afghanistan Asia & Oceania South Asia 2003  NA
## 6 Afghanistan Asia & Oceania South Asia 2004   0.161
## 7 Afghanistan Asia & Oceania South Asia 2005   0.127
## 8 Afghanistan Asia & Oceania South Asia 2006   0.104
## 9 Afghanistan Asia & Oceania South Asia 2007   0.119
```

Predicting spending

Predict 2020 spending $\rightarrow$ mean of spending (1999-2019)

Use loop to calculate means for all countries

```
## loop
pred.mean <- rep(NA,157)
c.names <- unique(spend_long$Country)
names(pred.mean) <- as.character(c.names)

for (i in 1:157){
  c.dat <- subset(spend_long, subset = (Country == c.names[i]))
  pred.mean[i] <- mean(c.dat$exp, na.rm = T)
}
```

Predicting spending for 2020

```
> pred.mean
        Afghanistan             Albania             Algeria              Angola           Argentina             Armenia
              7.694               4.804              11.679              11.421               2.865              15.727
          Australia             Austria          Azerbaijan             Bahrain          Bangladesh             Belarus
              5.117               1.622              11.593              13.654              10.249              30.557
            Belgium              Belize               Benin             Bolivia  Bosnia-Herzegovina            Botswana
              2.104               3.482               4.313               5.312               3.024               7.708
             Brazil              Brunei            Bulgaria        Burkina Faso             Burundi            Cambodia
              3.955               8.537               5.727               6.087              12.387               9.069
           Cameroon              Canada          Cape Verde Central African Rep.                Chad               Chile
              7.432               2.898               1.846              10.904              16.417              10.101
              China            Colombia   Congo, Dem. Rep.  Congo, Republic of          Costa Rica       Côte d'Ivoire
              8.148              11.338               9.083               8.326               0.000               7.180
            Croatia              Cyprus             Czechia             Denmark            Djibouti      Dominican Rep.
              4.204               4.972               3.230               2.517              15.135               4.516
            Ecuador               Egypt         El Salvador   Equatorial Guinea             Estonia            eSwatini
              7.901               6.539               4.408               5.625               4.614               6.041
           Ethiopia                Fiji             Finland              France               Gabon              Gambia
             10.330               5.670               2.705               3.599               7.089               3.736
            Georgia             Germany               Ghana              Greece           Guatemala              Guinea
             10.935               2.686               2.040               5.687               3.740              11.728
      Guinea-Bissau              Guyana               Haiti            Honduras             Hungary             Iceland
              9.553               4.377               0.001               4.366               2.512               0.000
              India           Indonesia                Iran                Iraq             Ireland              Israel
              9.693               4.122              14.319               6.366               1.472              14.203
              Italy             Jamaica               Japan              Jordan          Kazakhstan               Kenya
              3.099               2.672               2.560              15.356               4.723               6.172
       Korea, South              Kuwait          Kyrgyzstan                Laos              Latvia             Lebanon
             12.765              12.222               4.839               2.179               3.728              14.164
```

Good prediction?

Checking for errors:

```r
# Calculate errors & assign country names
errors <- mil_exp$`2020` - pred.mean
names(errors) <- c.names

# Average error
mean(errors, na.rm = T)
```

```
## [1] -0.01210775
```

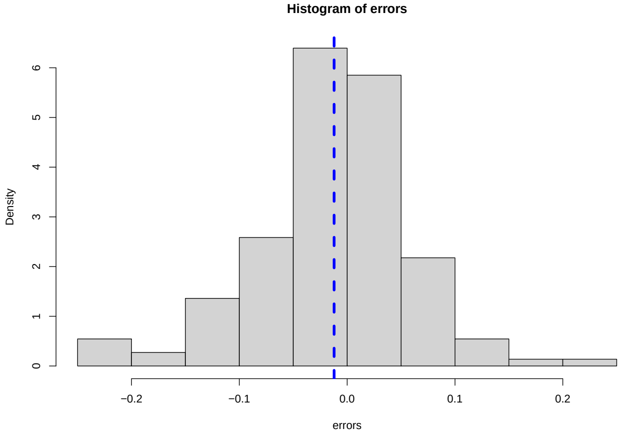
```r
# RMSE
sqrt(mean(errors^2, na.rm = T))
```

```
## [1] 0.07380063
```

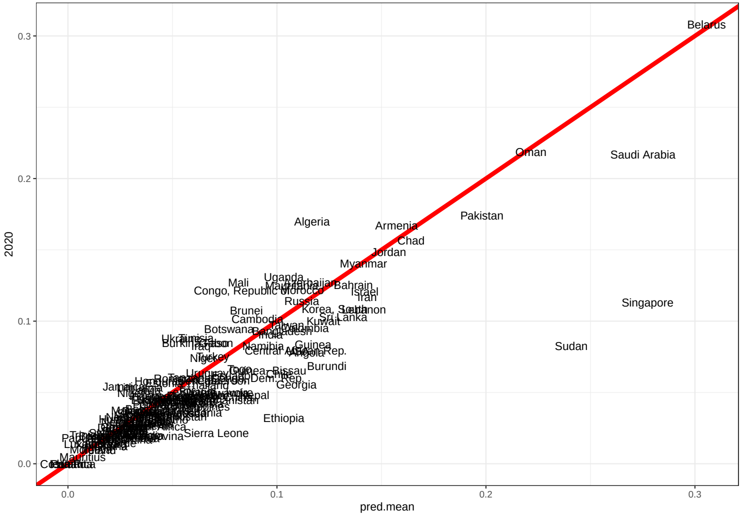
Prediction errors

How far off are we?

```
hist(errors, freq = FALSE)
abline(v = mean(errors, na.rm = T), lty = "dashed",
       col = "blue",lwd =4)
```



Accuracy of predictions



Find outlier predictions

Identify where we were off...

```r
# Errors distribution
summary(n.dat$error)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
## -0.164364 -0.017092 -0.004715 -0.008734  0.000374  0.053107        10
```

```r
# Create variable for large outliers
n.dat$large.inc <- NA
n.dat$large.inc[n.dat$error > 0.01] <- "Much More"
n.dat$large.inc[n.dat$error < -0.01] <- "Much Less"

# Create subset of outliers: less than average
n.dat2 <- n.dat %>%
  filter(large.inc == "Much Less") %>%
  mutate(error = error * 100) %>%
  select(Group1, error) %>% arrange(desc(error))

tail(n.dat2, n=7)
```

```
##                        Group1      error
## Sierra Leone           Africa  -4.945523
## Georgia                Europe  -5.375066
## Burundi                Africa  -5.521676
## Saudi Arabia      Middle East  -5.806989
## Ethiopia               Africa  -7.119952
## Sudan                  Africa -15.832405
## Singapore    Asia & Oceania -16.436356
```

Time series and predicted value

Focus on big-5 spenders

Format data to long-form

Create clear measure for expenditure

```
dat3 <- n.dat %>%
  filter(Country == "Russia" | Country == "USA" |
           Country == "China" | Country == "Iran" | Country == "Israel") %>%
  select(-Subgroup1, -error, -large.inc) %>%
  gather(year, exp, '1999':'2020',-Country, -Group1, -pred.mean) %>%
  arrange(Country) %>%
  mutate(exp = round(exp*100,2))
```

Working with dates

Working with dates:

- Package → library(lubridate)
- Define variables as dates and choose format
- We can calculate number of days between date variables

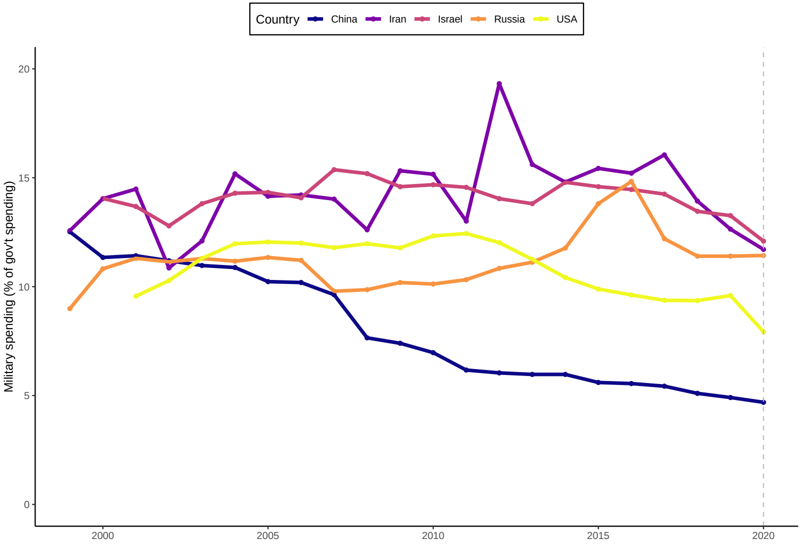
```
# Working with dates
arrive <- as.Date("2015-07-01")
today <- as.Date("2023-02-21")

# How long have I been in the US?
today - arrive
```

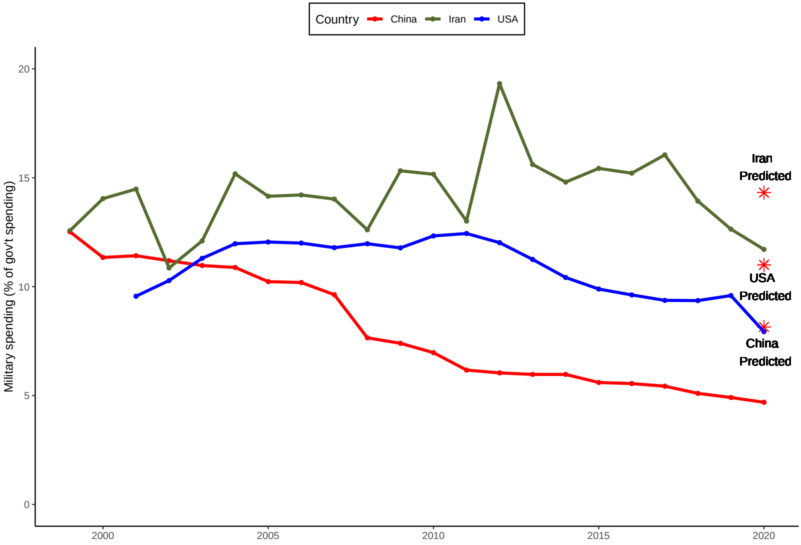
```
## Time difference of 2792 days
```

```
# Define dates in our expenditures data
dat3$year.f <- as.Date(dat3$year, format = "%Y")
dat3$year.f2 <- year(dat3$year.f)
```

Spending over time



Spending over time (and predicted 2020 - the 'big 3')



R Tech

Mastering R Markdown

- Use template.
- Code: ‘migrate’ from R script.
- Adding code chunks blocks.
- Text: captions, lists, bolded. . .
- Document organization - vertical spaces

US Military Aid

- Approximately $11-12 Billion per year.
- FP tool with various goals:
  - *quid-pro-quo* compliance with target government.
  - Augment US national security.
  - Require aid target cooperation.
- Outcomes? Not too promising...
  - Reduce cooperation (2011).
  - Reduce terrorism under certain conditions (2014).
  - Limited in lowering civil conflict (2018).
- Great data resource: *ForeignAssistance.gov* (Link)

Aid data

- US Aid (1990-2006)

```
# Explore Military aid data
dim(mil_aid2)
```

```
## [1] 2643   34
```

```
summary(mil_aid2$militaryaid)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00    0.00    0.20   34.49    1.30 3365.70       3
```

Predicting US Military Aid

- Predict 2006 levels → mean of aid (1990-2005)
- Use loop to calculate means for all countries

```
## Loop procedure
pred.aid <- rep(NA,168)
c.names <- unique(mil_aid2$country)
names(pred.aid) <- as.character(c.names)

for (i in 1:168){
  c.dat <- subset(mil_aid2, subset = (country == c.names[i]))
  pred.aid[i] <- mean(c.dat$militaryaid, na.rm = T)
}

pred.aid[pred.aid > 80]
```

```
##      Greece      Turkey        Iraq       Egypt      Jordan      Israel
##   196.29375   309.69375   179.95625  1595.04999   154.68125  2516.30624
## Afghanistan    Pakistan
##   115.82500    81.24375
```

Predicting Aid

- Check our predictions

```
# Error vectors and plot
aid.error <- mil_aid3$militaryaid - pred.aid
names(aid.error) <- c.names
mean(aid.error, na.rm = T)
```

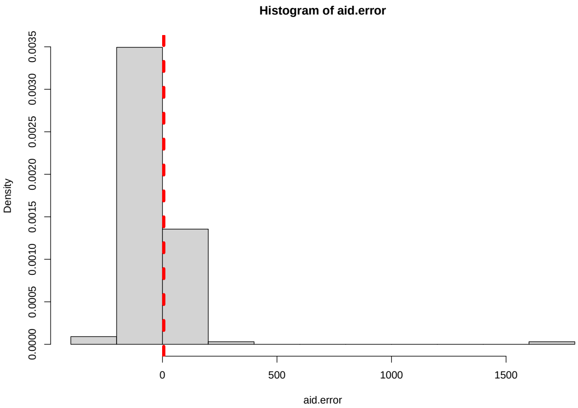
```
## [1] 5.719636
```

```
sqrt(mean(aid.error^2, na.rm = T))
```

```
## [1] 139.2933
```

Plot errors (outliers?)

```
hist(aid.error, freq = FALSE)
abline(v = mean(aid.error, na.rm = T), lty = "dashed", col = "red", lwd = 5)
```



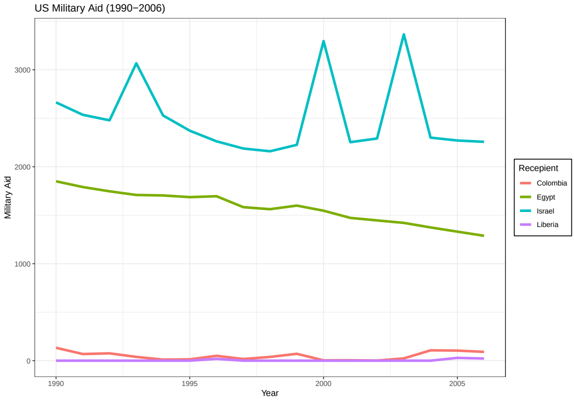

```
aid.error[aid.error > 1000]
```

```
##        <NA>        <NA> Afghanistan
##          NA          NA    1691.175
```

US Military aid: Time trends

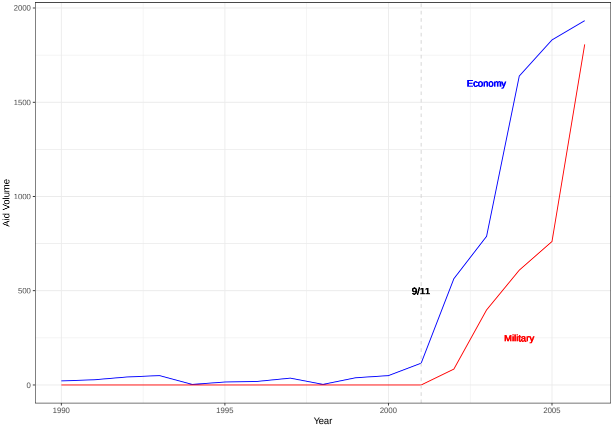
```r
mil_aid4 <- mil_aid %>%
  filter(country == "Colombia" | country == "Egypt" | country == "Israel" | country == "Liberia")

ggplot(mil_aid4, aes(x = year, y = militaryaid)) +
  geom_line(aes(color = country), size = 1.5) +
  scale_color_discrete(name = "Recepient") +
  theme_bw() + xlab("Year") + ylab("Military Aid") + ggtitle("US Military Aid (1990-2006)") +
  theme(legend.position = "right",
        legend.background = element_rect(size = 0.5, linetype = "solid", colour = "black"))
```



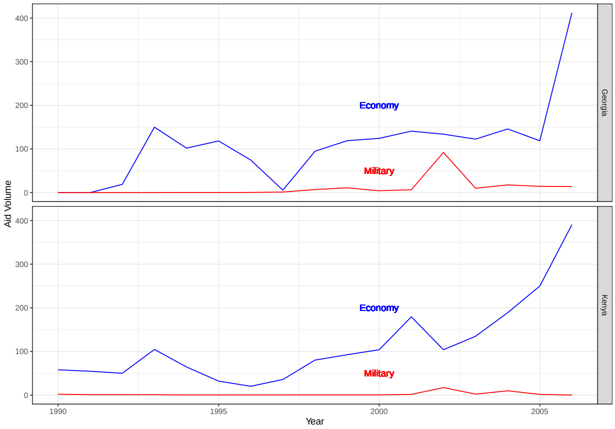
Military and Economic aid: Afghanistan (1990-2006)

```r
mil_aid %>%
  filter(country == "Afghanistan") %>%
  ggplot() +
  geom_line(aes(year,economicaid), color = "blue") + xlab("Year") +
  geom_line(aes(year,militaryaid), color = "red") + ylab("Aid Volume") +
  geom_text(aes(x = 2003, y = 1600, label = "Economy"), color = "blue") +
  geom_text(aes(x = 2004, y = 250, label = "Military"), color = "red") +
  geom_vline(aes(xintercept = 2001), linetype = "dashed", color = "lightgrey") +
  geom_text(aes(x = 2001, y = 500, label = "9/11"), color = "black") + theme_bw()
```



Military and Econ aid: Always tracking??

```
mil_aid %>%
  filter(country == "Georgia" | country == "Kenya") %>%
  ggplot(aes(group = country)) +
  geom_line(aes(year,economicaid), color = "blue") + xlab("Year") +
  geom_line(aes(year,militaryaid), color = "red") + ylab("Aid Volume") +
  geom_text(aes(x = 2000, y = 200, label = "Economy"), color = "blue") +
  geom_text(aes(x = 2000, y = 50, label = "Military"), color = "red") +
  facet_grid(country~.) + theme_bw()
```



Military and Economic aid (1990-2006)

- Checking for correlations

```
# Build data frame for means of aid types
type <- c("Military","Economic")
value <- c(mean(mil_aid$militaryaid,na.rm = T),
           mean(mil_aid$economicaid,na.rm = T))
aid_types <- data.frame(type,value)
aid_types
```

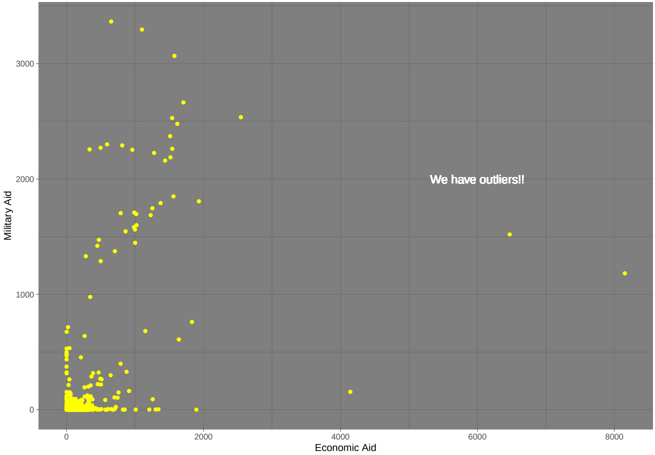
```
##       type    value
## 1 Military 33.08976
## 2 Economic 66.11048
```

```
# Correlation
cor(mil_aid$militaryaid, mil_aid$economicaid, use = "complete.obs")
```

```
## [1] 0.5559843
```

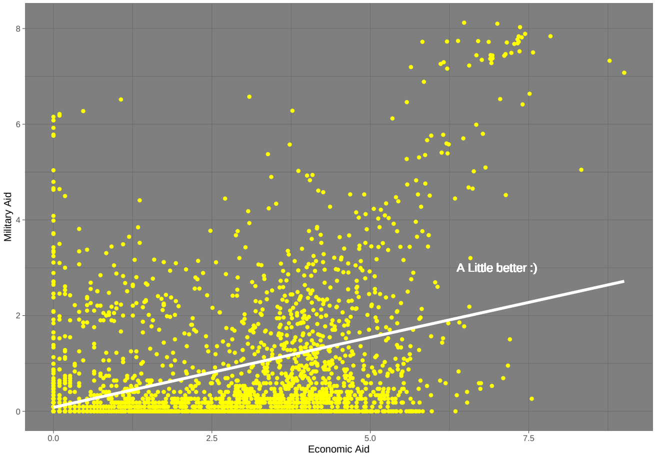
Plotting corrleation

```
ggplot(mil_aid, aes(x=economicaid, y=militaryaid)) +
  geom_point(color = "yellow") +
  xlab("Economic Aid") + ylab("Military Aid") +
  geom_text(aes(x =6000, y = 2000, label = "We have outliers!!"), color = "white", size = 4.5) +
  theme_dark()
```



Plotting correlations: “Remove” outliers

```
ggplot(mil_aid, aes(x=logeconomicaid, y=logmilitaryaid)) +
  geom_point(color = "yellow") +
  geom_smooth(method = "lm", se = F, color = "white", size = 1.5) +
  xlab("Economic Aid") + ylab("Military Aid") +
  geom_text(aes(x =7, y = 3, label = "A Little better :)"), color = "white", size = 4.5) +
  theme_dark()
```



Wrapping up week 6

Summary:

- Predictions...
- Using data to find the 'best- guess' of some quantity.
- Repeated computations? Use Loops.
- Always check for prediction errors.
- Classification errors: false positive and false negative.
- Data over time
- US military aid data: predictions, errors and some insights