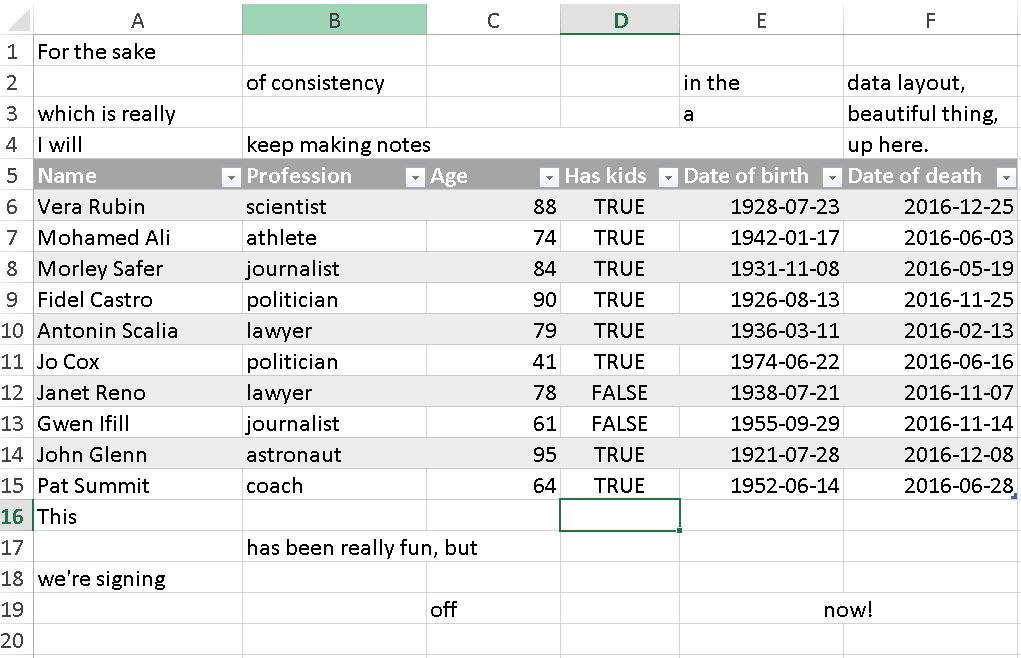
Graduate Seminar

# Excel Import Demo

## Deaths Data set



* Environment Tab --> Import Dataset

## First attempt

library(readxl)  
deaths <- read\_excel("~/Graduate\_Seminar\_Presentation/deaths.xlsx")  
  
deaths

## # A tibble: 18 x 6  
## `Lots of people` X\_\_1 X\_\_2 X\_\_3  
## <chr> <chr> <chr> <chr>  
## 1 simply cannot resist writing <NA> <NA> <NA>  
## 2 at the top <NA>  
## 3 or merging <NA> <NA>  
## 4 Name Profession Age Has kids  
## 5 David Bowie musician 69 TRUE  
## 6 Carrie Fisher actor 60 TRUE  
## 7 Chuck Berry musician 90 TRUE  
## 8 Bill Paxton actor 61 TRUE  
## 9 Prince musician 57 TRUE  
## 10 Alan Rickman actor 69 FALSE  
## 11 Florence Henderson actor 82 TRUE  
## 12 Harper Lee author 89 FALSE  
## 13 Zsa Zsa Gábor actor 99 TRUE  
## 14 George Michael musician 53 FALSE  
## 15 Some <NA> <NA> <NA>  
## 16 <NA> also like to write stuff <NA> <NA>  
## 17 <NA> <NA> at the bottom,  
## 18 <NA> <NA> <NA> <NA>  
## # ... with 2 more variables: X\_\_4 <chr>, X\_\_5 <chr>

## Second (& successful) attempt

library(readxl)  
deaths <- read\_excel("~/Graduate\_Seminar\_Presentation/deaths.xlsx",   
 range = cell\_rows(5:15))  
  
deaths

## # A tibble: 10 x 6  
## Name Profession Age `Has kids` `Date of birth`  
## <chr> <chr> <dbl> <lgl> <dttm>  
## 1 David Bowie musician 69 TRUE 1947-01-08  
## 2 Carrie Fisher actor 60 TRUE 1956-10-21  
## 3 Chuck Berry musician 90 TRUE 1926-10-18  
## 4 Bill Paxton actor 61 TRUE 1955-05-17  
## 5 Prince musician 57 TRUE 1958-06-07  
## 6 Alan Rickman actor 69 FALSE 1946-02-21  
## 7 Florence Henderson actor 82 TRUE 1934-02-14  
## 8 Harper Lee author 89 FALSE 1926-04-28  
## 9 Zsa Zsa Gábor actor 99 TRUE 1917-02-06  
## 10 George Michael musician 53 FALSE 1963-06-25  
## # ... with 1 more variables: `Date of death` <dttm>

# Data Munging Example

## Pipe Operator and mutate

Create a new column: birthplace

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

deaths %>%   
 mutate(birthplace = c("UK", "US", "US", "US", "US",   
 "UK", "US", "US", "Hungary",  
 "UK"))

## # A tibble: 10 x 7  
## Name Profession Age `Has kids` `Date of birth`  
## <chr> <chr> <dbl> <lgl> <dttm>  
## 1 David Bowie musician 69 TRUE 1947-01-08  
## 2 Carrie Fisher actor 60 TRUE 1956-10-21  
## 3 Chuck Berry musician 90 TRUE 1926-10-18  
## 4 Bill Paxton actor 61 TRUE 1955-05-17  
## 5 Prince musician 57 TRUE 1958-06-07  
## 6 Alan Rickman actor 69 FALSE 1946-02-21  
## 7 Florence Henderson actor 82 TRUE 1934-02-14  
## 8 Harper Lee author 89 FALSE 1926-04-28  
## 9 Zsa Zsa Gábor actor 99 TRUE 1917-02-06  
## 10 George Michael musician 53 FALSE 1963-06-25  
## # ... with 2 more variables: `Date of death` <dttm>, birthplace <chr>

# Data Exploration

## Gapminder Data

library(gapminder)  
  
gapminder

## # A tibble: 1,704 x 6  
## country continent year lifeExp pop gdpPercap  
## <fctr> <fctr> <int> <dbl> <int> <dbl>  
## 1 Afghanistan Asia 1952 28.801 8425333 779.4453  
## 2 Afghanistan Asia 1957 30.332 9240934 820.8530  
## 3 Afghanistan Asia 1962 31.997 10267083 853.1007  
## 4 Afghanistan Asia 1967 34.020 11537966 836.1971  
## 5 Afghanistan Asia 1972 36.088 13079460 739.9811  
## 6 Afghanistan Asia 1977 38.438 14880372 786.1134  
## 7 Afghanistan Asia 1982 39.854 12881816 978.0114  
## 8 Afghanistan Asia 1987 40.822 13867957 852.3959  
## 9 Afghanistan Asia 1992 41.674 16317921 649.3414  
## 10 Afghanistan Asia 1997 41.763 22227415 635.3414  
## # ... with 1,694 more rows

## Summary Statistics

### All countries

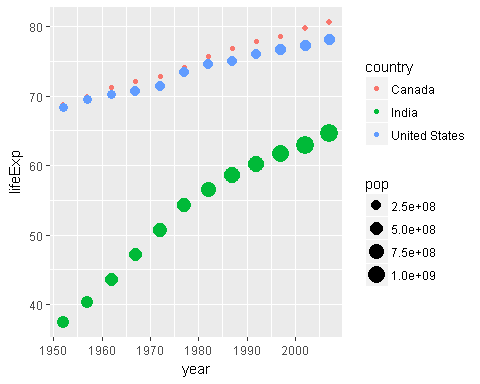
gapminder %>%   
 group\_by(country) %>%  
 select(lifeExp, pop, gdpPercap) %>%   
 summarize\_all(funs(min, max, mean, sd, median))

## Adding missing grouping variables: `country`

## # A tibble: 142 x 16  
## country lifeExp\_min pop\_min gdpPercap\_min lifeExp\_max pop\_max  
## <fctr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 Afghanistan 28.801 8425333 635.3414 43.828 31889923  
## 2 Albania 55.230 1282697 1601.0561 76.423 3600523  
## 3 Algeria 43.077 9279525 2449.0082 72.301 33333216  
## 4 Angola 30.015 4232095 2277.1409 42.731 12420476  
## 5 Argentina 62.485 17876956 5911.3151 75.320 40301927  
## 6 Australia 69.120 8691212 10039.5956 81.235 20434176  
## 7 Austria 66.800 6927772 6137.0765 79.829 8199783  
## 8 Bahrain 50.939 120447 9867.0848 75.635 708573  
## 9 Bangladesh 37.484 46886859 630.2336 64.062 150448339  
## 10 Belgium 68.000 8730405 8343.1051 79.441 10392226  
## # ... with 132 more rows, and 10 more variables: gdpPercap\_max <dbl>,  
## # lifeExp\_mean <dbl>, pop\_mean <dbl>, gdpPercap\_mean <dbl>,  
## # lifeExp\_sd <dbl>, pop\_sd <dbl>, gdpPercap\_sd <dbl>,  
## # lifeExp\_median <dbl>, pop\_median <dbl>, gdpPercap\_median <dbl>

#### Plot (Interactive and non-interactive)

library(ggplot2)  
gapminder %>%   
 filter(country %in% c("Canada", "United States", "India")) %>%   
 ggplot(data = .) +  
 geom\_point(aes(x = year, y = lifeExp, size = pop, color = country))



# gapminder %>%   
# filter(country %in% c("Canada", "United States", "India")) %>%   
# ggplot(data = .) +  
# geom\_point(aes(x = year, y = lifeExp, size = pop)) +  
# facet\_wrap(~country)

library(plotly)

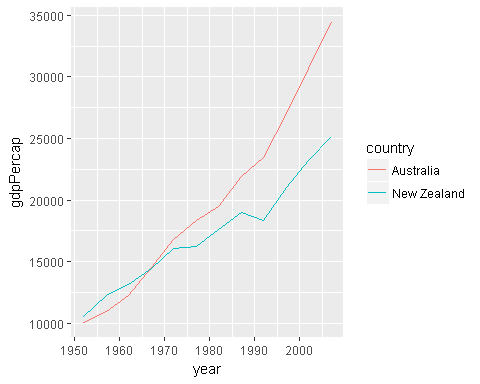
##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

## The following object is masked from 'package:graphics':  
##   
## layout

ggplot(data = gapminder %>%   
 filter(continent=="Oceania")) + #Europe Africa  
 geom\_line(aes(x = year, y = gdpPercap, color = country))



ggplotly()

## We recommend that you use the dev version of ggplot2 with `ggplotly()`  
## Install it with: `devtools::install\_github('hadley/ggplot2')`

# Modelling

## Gapminder data (Nested)

library(tidyr)  
library(purrr)

##   
## Attaching package: 'purrr'

## The following objects are masked from 'package:dplyr':  
##   
## contains, order\_by

by\_country <- gapminder %>%   
 group\_by(country, continent) %>%   
 nest()  
by\_country

## # A tibble: 142 x 3  
## country continent data  
## <fctr> <fctr> <list>  
## 1 Afghanistan Asia <tibble [12 x 4]>  
## 2 Albania Europe <tibble [12 x 4]>  
## 3 Algeria Africa <tibble [12 x 4]>  
## 4 Angola Africa <tibble [12 x 4]>  
## 5 Argentina Americas <tibble [12 x 4]>  
## 6 Australia Oceania <tibble [12 x 4]>  
## 7 Austria Europe <tibble [12 x 4]>  
## 8 Bahrain Asia <tibble [12 x 4]>  
## 9 Bangladesh Asia <tibble [12 x 4]>  
## 10 Belgium Europe <tibble [12 x 4]>  
## # ... with 132 more rows

## Fitting Models:

#Model  
country\_model <- function(df) {  
 lm(lifeExp ~ year, data = df)  
}  
  
# Fitting model  
by\_country <- by\_country %>%   
 mutate(model = map(data, country\_model))  
by\_country

## # A tibble: 142 x 4  
## country continent data model  
## <fctr> <fctr> <list> <list>  
## 1 Afghanistan Asia <tibble [12 x 4]> <S3: lm>  
## 2 Albania Europe <tibble [12 x 4]> <S3: lm>  
## 3 Algeria Africa <tibble [12 x 4]> <S3: lm>  
## 4 Angola Africa <tibble [12 x 4]> <S3: lm>  
## 5 Argentina Americas <tibble [12 x 4]> <S3: lm>  
## 6 Australia Oceania <tibble [12 x 4]> <S3: lm>  
## 7 Austria Europe <tibble [12 x 4]> <S3: lm>  
## 8 Bahrain Asia <tibble [12 x 4]> <S3: lm>  
## 9 Bangladesh Asia <tibble [12 x 4]> <S3: lm>  
## 10 Belgium Europe <tibble [12 x 4]> <S3: lm>  
## # ... with 132 more rows

# Getting Goodness of fit and Significance

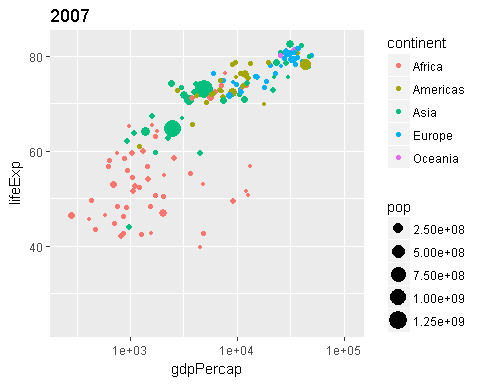
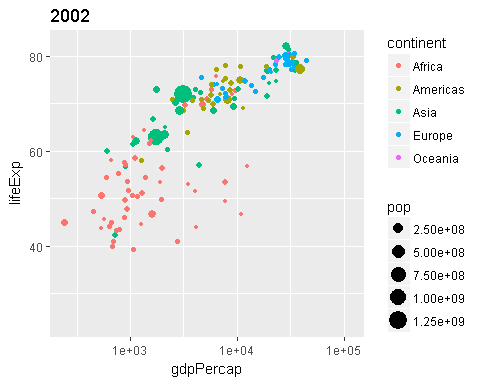
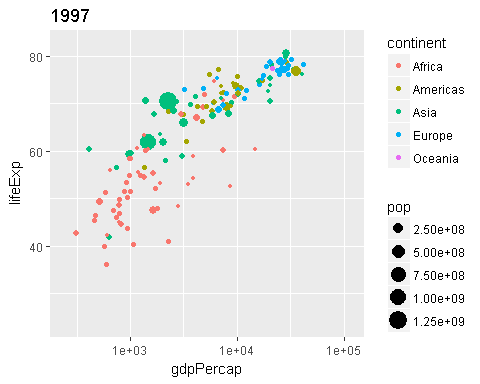
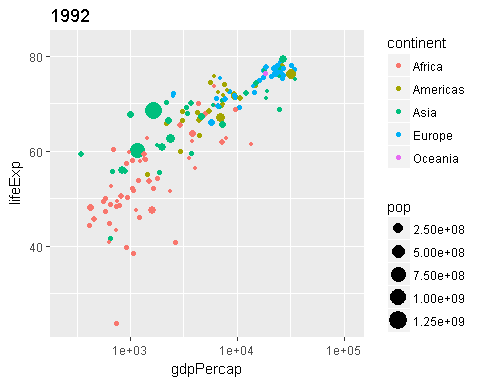
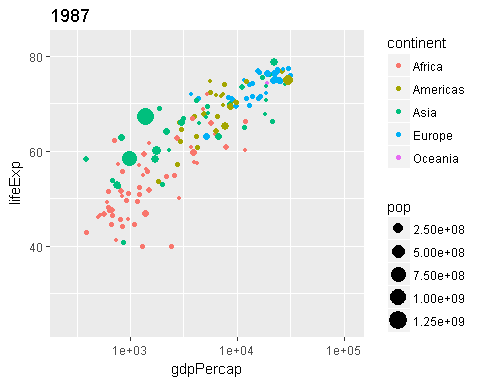
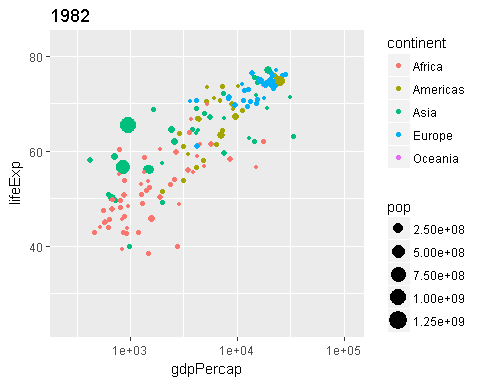
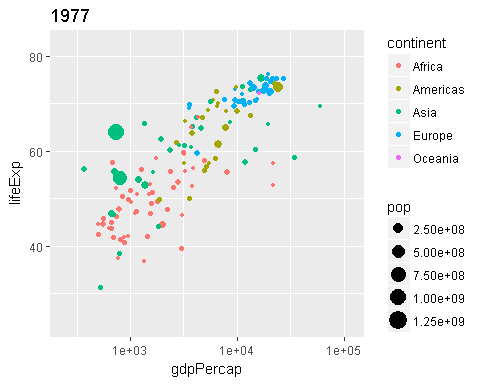
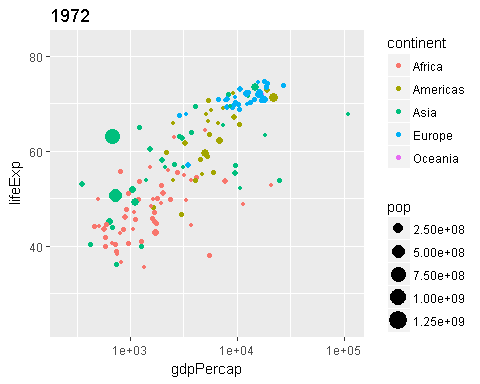
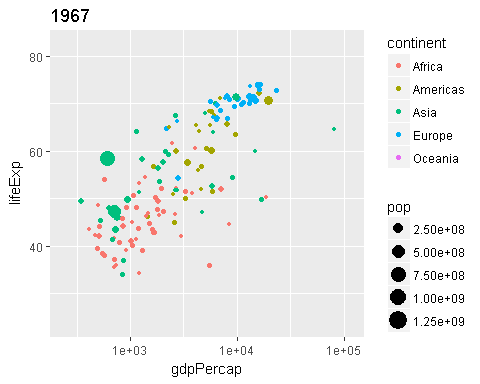
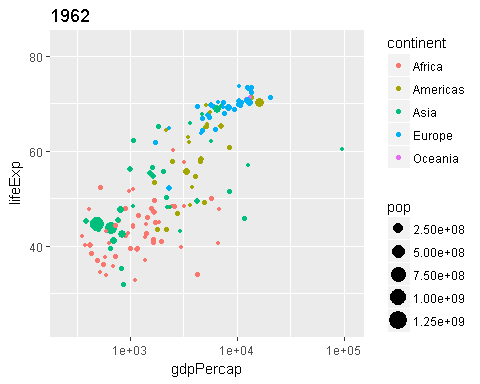
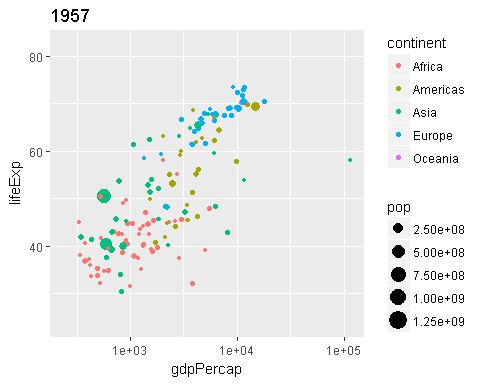
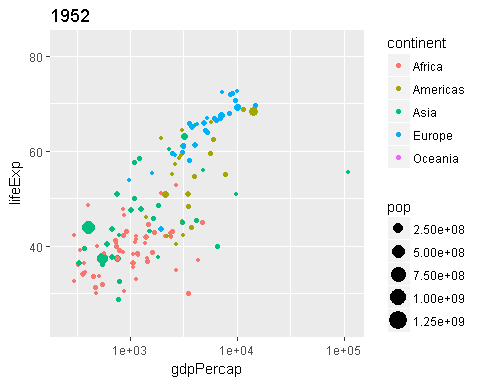
library(broom)  
by\_country %>%   
 mutate(glance = map(model, glance)) %>%   
 unnest(glance, .drop = TRUE)

## # A tibble: 142 x 13  
## country continent r.squared adj.r.squared sigma statistic  
## <fctr> <fctr> <dbl> <dbl> <dbl> <dbl>  
## 1 Afghanistan Asia 0.9477123 0.9424835 1.2227880 181.24941  
## 2 Albania Europe 0.9105778 0.9016355 1.9830615 101.82901  
## 3 Algeria Africa 0.9851172 0.9836289 1.3230064 661.91709  
## 4 Angola Africa 0.8878146 0.8765961 1.4070091 79.13818  
## 5 Argentina Americas 0.9955681 0.9951249 0.2923072 2246.36635  
## 6 Australia Oceania 0.9796477 0.9776125 0.6206086 481.34586  
## 7 Austria Europe 0.9921340 0.9913474 0.4074094 1261.29629  
## 8 Bahrain Asia 0.9667398 0.9634138 1.6395865 290.65974  
## 9 Bangladesh Asia 0.9893609 0.9882970 0.9766908 929.92637  
## 10 Belgium Europe 0.9945406 0.9939946 0.2929025 1821.68840  
## # ... with 132 more rows, and 7 more variables: p.value <dbl>, df <int>,  
## # logLik <dbl>, AIC <dbl>, BIC <dbl>, deviance <dbl>, df.residual <int>

# Communication

## Gapminder animation

library(gganimate)  
  
p <- ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, color = continent, frame = year)) +  
 geom\_point() +  
 scale\_x\_log10()  
gganimate(p)



sessionInfo()

## R version 3.4.1 (2017-06-30)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 8.1 x64 (build 9600)  
##   
## Matrix products: default  
##   
## locale:  
## [1] LC\_COLLATE=English\_Canada.1252 LC\_CTYPE=English\_Canada.1252   
## [3] LC\_MONETARY=English\_Canada.1252 LC\_NUMERIC=C   
## [5] LC\_TIME=English\_Canada.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] gganimate\_0.1.0.9000 broom\_0.4.2 purrr\_0.2.2.2   
## [4] tidyr\_0.6.3 plotly\_4.7.0 ggplot2\_2.2.1   
## [7] gapminder\_0.2.0 bindrcpp\_0.2 dplyr\_0.7.2   
## [10] readxl\_1.0.0   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.12 cellranger\_1.1.0 compiler\_3.4.1   
## [4] plyr\_1.8.4 bindr\_0.1 tools\_3.4.1   
## [7] digest\_0.6.12 lattice\_0.20-35 nlme\_3.1-131   
## [10] jsonlite\_1.5 evaluate\_0.10.1 tibble\_1.3.3   
## [13] gtable\_0.2.0 viridisLite\_0.2.0 pkgconfig\_2.0.1   
## [16] rlang\_0.1.1 psych\_1.7.5 shiny\_1.0.3   
## [19] crosstalk\_1.0.0 parallel\_3.4.1 yaml\_2.1.14   
## [22] stringr\_1.2.0 httr\_1.2.1 knitr\_1.16   
## [25] htmlwidgets\_0.9 rprojroot\_1.2 grid\_3.4.1   
## [28] glue\_1.1.1 data.table\_1.10.4 R6\_2.2.2   
## [31] foreign\_0.8-69 rmarkdown\_1.6 reshape2\_1.4.2   
## [34] magrittr\_1.5 backports\_1.1.0 scales\_0.4.1   
## [37] htmltools\_0.3.6 mnormt\_1.5-5 assertthat\_0.2.0   
## [40] xtable\_1.8-2 mime\_0.5 colorspace\_1.3-2   
## [43] httpuv\_1.3.5 labeling\_0.3 stringi\_1.1.5   
## [46] lazyeval\_0.2.0 munsell\_0.4.3