



SOCIETY OF
ACTUARIES®

2019 **ANNUAL
MEETING**
& EXHIBIT

October 27-30
Toronto, Canada

Session 174: R 101: Available Actuarial Packages & Creating Reproducible Actuarial Analyses

[SOA Antitrust Compliance Guidelines](#)

[SOA Presentation Disclaimer](#)

Session 174: R 101: Available Actuarial Packages & Creating Reproducible Actuarial Analyses

Ron Curran, FSA, MAAA

Hadrien Dykiel

Nicholas Hanewinckel, FSA, CERA

Matthew Zhang, FSA, MAAA, CERA

October 30, 2019



SOCIETY OF ACTUARIES

Antitrust Compliance Guidelines

Active participation in the Society of Actuaries is an important aspect of membership. While the positive contributions of professional societies and associations are well-recognized and encouraged, association activities are vulnerable to close antitrust scrutiny. By their very nature, associations bring together industry competitors and other market participants.

The United States antitrust laws aim to protect consumers by preserving the free economy and prohibiting anti-competitive business practices; they promote competition. There are both state and federal antitrust laws, although state antitrust laws closely follow federal law. The Sherman Act, is the primary U.S. antitrust law pertaining to association activities. The Sherman Act prohibits every contract, combination or conspiracy that places an unreasonable restraint on trade. There are, however, some activities that are illegal under all circumstances, such as price fixing, market allocation and collusive bidding.

There is no safe harbor under the antitrust law for professional association activities. Therefore, association meeting participants should refrain from discussing any activity that could potentially be construed as having an anti-competitive effect. Discussions relating to product or service pricing, market allocations, membership restrictions, product standardization or other conditions on trade could arguably be perceived as a restraint on trade and may expose the SOA and its members to antitrust enforcement procedures.

While participating in all SOA in person meetings, webinars, teleconferences or side discussions, you should avoid discussing competitively sensitive information with competitors and follow these guidelines:

- **Do not** discuss prices for services or products or anything else that might affect prices
- **Do not** discuss what you or other entities plan to do in a particular geographic or product markets or with particular customers.
- **Do not** speak on behalf of the SOA or any of its committees unless specifically authorized to do so.
- **Do** leave a meeting where any anticompetitive pricing or market allocation discussion occurs.
- **Do** alert SOA staff and/or legal counsel to any concerning discussions
- **Do** consult with legal counsel before raising any matter or making a statement that may involve competitively sensitive information.

Adherence to these guidelines involves not only avoidance of antitrust violations, but avoidance of behavior which might be so construed. These guidelines only provide an overview of prohibited activities. SOA legal counsel reviews meeting agenda and materials as deemed appropriate and any discussion that departs from the formal agenda should be scrutinized carefully. Antitrust compliance is everyone's responsibility; however, please seek legal counsel if you have any questions or concerns.

Presentation Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the participants individually and, unless expressly stated to the contrary, are not the opinion or position of the Society of Actuaries, its cosponsors or its committees. The Society of Actuaries does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented. Attendees should note that the sessions are audio-recorded and may be published in various media, including print, audio and video formats without further notice.



Session Presented By:

Predictive Analytics and Futurism Section

Provides opportunities for actuaries to deepen their understanding of predictive analytics and emerging technologies relevant to the future of the actuarial profession and insurance industry.

Section Developed Content & Benefits



Predictive Analytics and Futurism Newsletter

Discusses futurism and the latest predictive analytics trends. Published three times a year. Digital editions now available.



SOA Meetings and Seminars

Section developed content presented during meeting sessions and seminars.



Podcasts

Expert led technical podcasts exploring the latest predictive analytics concepts and techniques.



Webcasts

Discounts on section developed webcasts. Free access to section created webcasts over one-year old.

Join the PAF Section Today! SOA.org/PAF



CONNECT WITH SECTION MEMBERS

Moderator and Presenters

- Ron Curran, FSA, MAAA
 - Assistant Vice President and Actuary at Hannover Re
 - Leads the corporate economic capital reporting team of the ERM department.
 - Utilizes the latest machine learning tools in R to run stochastic proxy models.
- Hadrien Dykiel, Customer Success at RStudio
 - Helps RStudio's insurance customers deploy and leverage R at scale in the enterprise.
 - Works with both large and small insurance companies to architect their RStudio infrastructure and explore the different ways R can be used to solve new problems and optimize existing workflows for both actuaries and other insurance professionals.
- Nick Hanewinckel, FSA, CERA
 - Assistant Vice President and Actuary at Hannover Re
 - Worked for several years in Predictive Modelling in roles related to everything from underwriting to rate setting.
 - Worked in R, Python, and Spark.
- Matthew Zhang, FSA, MAAA, CERA
 - Consultant at Oliver Wyman
 - Provides actuarial consulting services to various insurance entities and organizations.
 - Joined Oliver Wyman in 2017 after having previously worked at FIS (formerly SunGard), supporting clients with model implementation, maintenance, governance, and optimization within the Prophet suite. Prior to that, he worked for Ernst & Young, primarily focused on model validation and actuarial audit. Much of his experience is involved with building, modifying, or reviewing actuarial models.

R for Insurance 101

Available Actuarial Packages & Creating Reproducible
Actuarial Analyses

Hadrien Dykiel

Customer Success @ RStudio



Agenda

- What is R?
 - Open Source Software
 - Benefits
 - Overview of R ecosystem
- How can it help you?
 - Useful packages
 - Real world insurance examples

What is R?

- R is a powerful programming language frequently used for data analysis, prediction modeling, and data visualization.



R is Open Source

Open Source Software (OSS) generally means that the source code is made available with a license which provides the rights to:

- > Examine
- > Modify
- > Redistribute



Benefits of OSS like R:

- Powerful (14,924+ Packages on CRAN!)
- Inexpensive (R is free)
- Speed to Market (use the latest & greatest libraries)
- Security (R is transparent)
- Community (it's friendly)

R code can be “packaged” and shared with others



```
ggplot()  
filter()      mutate()  
write_csv()
```

R Language

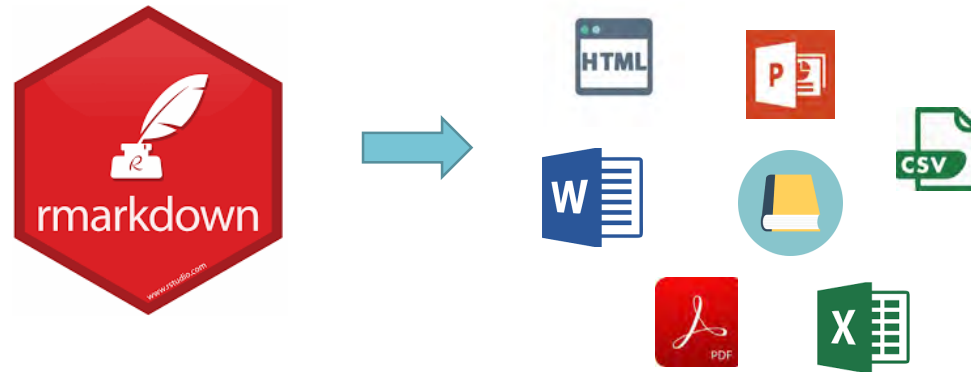
R packages

functions

R can be used to create a lot of Data Products



rmarkdown Demo



In summary, R is...

- A powerful tool
- Becoming easier to learn
- Gaining more & more momentum in the insurance industry
- Useful for data wrangling, analysis, visualization, reproducible reporting, interactive applications, APIs

SOCIETY OF ACTUARIES

Antitrust Compliance Guidelines

Active participation in the Society of Actuaries is an important aspect of membership. While the positive contributions of professional societies and associations are well-recognized and encouraged, association activities are vulnerable to close antitrust scrutiny. By their very nature, associations bring together industry competitors and other market participants.

The United States antitrust laws aim to protect consumers by preserving the free economy and prohibiting anti-competitive business practices; they promote competition. There are both state and federal antitrust laws, although state antitrust laws closely follow federal law. The Sherman Act, is the primary U.S. antitrust law pertaining to association activities. The Sherman Act prohibits every contract, combination or conspiracy that places an unreasonable restraint on trade. There are, however, some activities that are illegal under all circumstances, such as price fixing, market allocation and collusive bidding.

There is no safe harbor under the antitrust law for professional association activities. Therefore, association meeting participants should refrain from discussing any activity that could potentially be construed as having an anti-competitive effect. Discussions relating to product or service pricing, market allocations, membership restrictions, product standardization or other conditions on trade could arguably be perceived as a restraint on trade and may expose the SOA and its members to antitrust enforcement procedures.

While participating in all SOA in person meetings, webinars, teleconferences or side discussions, you should avoid discussing competitively sensitive information with competitors and follow these guidelines:

- **Do not** discuss prices for services or products or anything else that might affect prices
- **Do not** discuss what you or other entities plan to do in a particular geographic or product markets or with particular customers.
- **Do not** speak on behalf of the SOA or any of its committees unless specifically authorized to do so.
- **Do** leave a meeting where any anticompetitive pricing or market allocation discussion occurs.
- **Do** alert SOA staff and/or legal counsel to any concerning discussions
- **Do** consult with legal counsel before raising any matter or making a statement that may involve competitively sensitive information.

Adherence to these guidelines involves not only avoidance of antitrust violations, but avoidance of behavior which might be so construed. These guidelines only provide an overview of prohibited activities. SOA legal counsel reviews meeting agenda and materials as deemed appropriate and any discussion that departs from the formal agenda should be scrutinized carefully. Antitrust compliance is everyone's responsibility; however, please seek legal counsel if you have any questions or concerns.

Presentation Disclaimer

Presentations are intended for educational purposes only and do not replace independent professional judgment. Statements of fact and opinions expressed are those of the participants individually and, unless expressly stated to the contrary, are not the opinion or position of the Society of Actuaries, its cosponsors or its committees. The Society of Actuaries does not endorse or approve, and assumes no responsibility for, the content, accuracy or completeness of the information presented. Attendees should note that the sessions are audio-recorded and may be published in various media, including print, audio and video formats without further notice.



*somewhat
different*



R 101

Available Actuarial Packages & Creating Reproducible Actuarial Analyses

Nick Hanewinckel, FSA, CERA
AVP and Actuary , HannoverRe
SOA Annual Meeting 2009 - 2019-10-30 - Session 174

hannover **re**[®]

Disclaimer

The information provided in this presentation does in no way whatsoever constitute legal, accounting, tax or other professional advice.

While Hannover Rück SE has endeavoured to include in this presentation information it believes to be reliable, complete and up-to-date, the company does not make any representation or warranty, express or implied, as to the accuracy, completeness or updated status of such information.

Therefore, in no case whatsoever will Hannover Rück SE and its affiliated companies or directors, officers or employees be liable to anyone for any decision made or action taken in conjunction with the information in this presentation or for any related damages.

© Hannover Rück SE. All rights reserved.

Hannover Re is the registered service mark of Hannover Rück SE

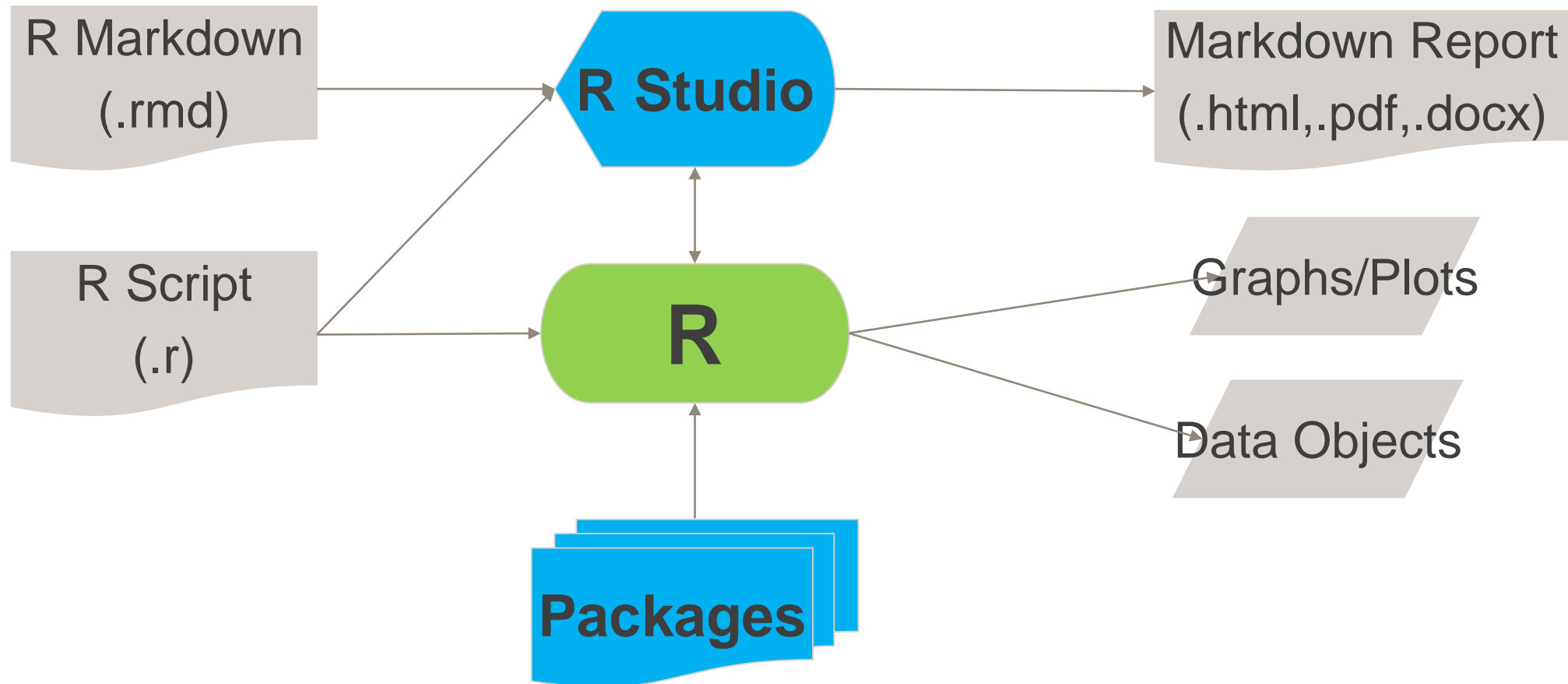
*somewhat
different*

Basics

hannover re[®]

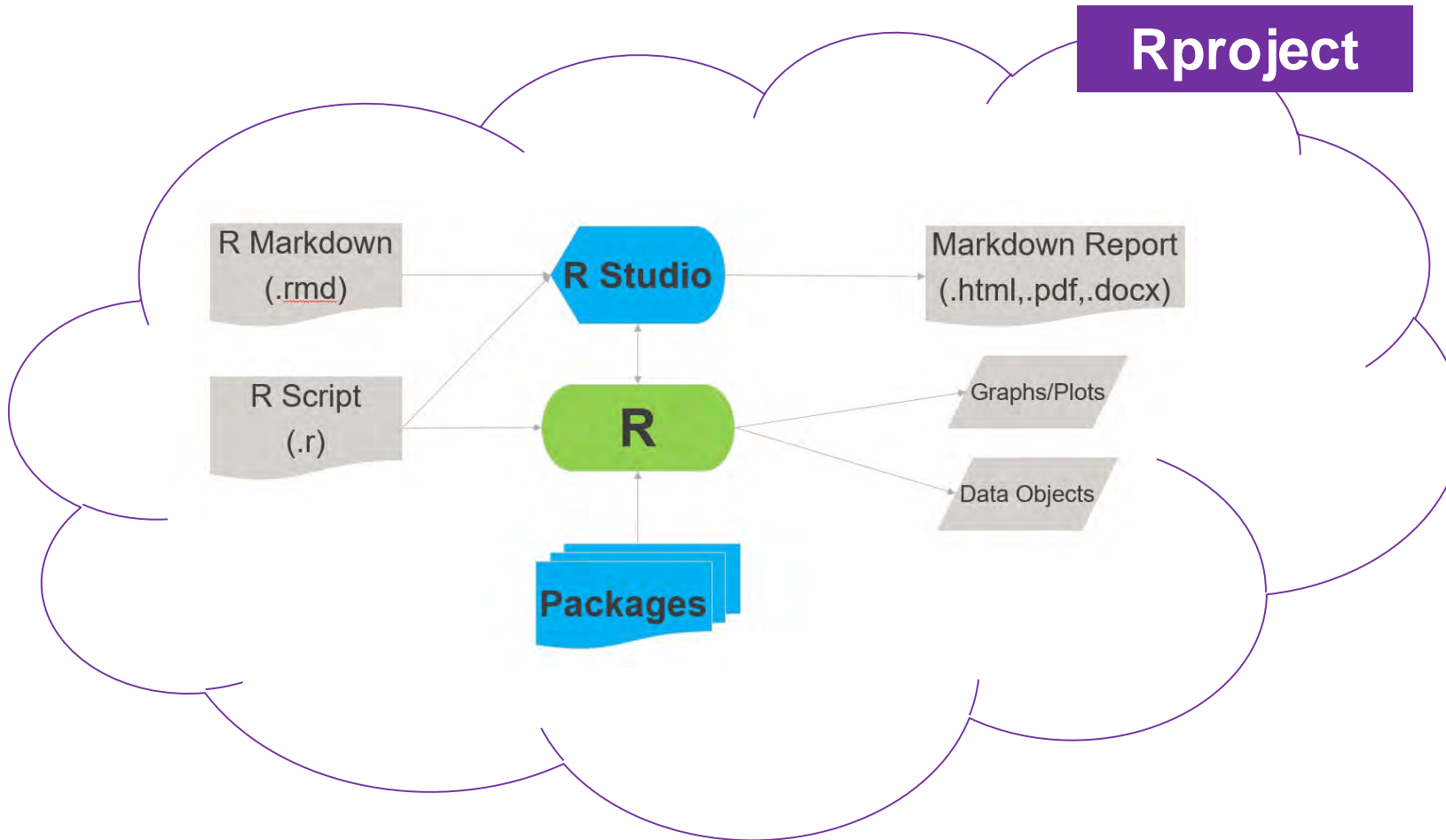
Multiple Tools to Make R Work

*somewhat
different*



Optional: Rproject

*somewhat
different*



- Organizes Files
- References a single directory
- Movable
- Usable by Multiple Users
- Helps rmarkdown deal with a “working directory”

The screenshot shows the RStudio interface with three panels highlighted in blue: 'Environment', 'Scripts', and 'Console'. The 'Environment' panel on the right lists objects like 'DSOR', 'dat', 'hmd', 'JAMS_F', 'JAMS_M', 'smt', and 'ssa'. The 'Scripts' panel in the center shows R code for plotting and model predictions. The 'Console' panel at the bottom shows the execution of the code.

```
106 * Scales="free", capt=NULL,...){
107   tmp <- tblSinglePred(regressor, vFacet, hFacet, color, ...)
108   Age <- range(tmp$AttainedAge)
109   if(is.null(color)){
110     a <- ggplot(tmp, aes_(x=as.name(regressor), y=quote(qxModel)))+
111       geom_line()+
112       facet_grid(paste0(vFacet, "~", hFacet), scales=Scales)+
113       labs(title=paste0("Attained Age: ", Age[1], ifelse(Age[2]>Age[1], paste0(" - ", Age[2]), "")),
114            subtitle=capt)
115     b <- ggplot(tmp, aes_(x=as.name(regressor), y=quote(ssaMultiple)))+
116       geom_line()+
117       facet_grid(paste0(vFacet, "~", hFacet), scales=Scales)+
118       labs(title=paste0("Attained Age: ", Age[1], ifelse(Age[2]>Age[1], paste0(" - ", Age[2]), "")),
119            subtitle=capt)
120   }else{
121     tmp[,eval(color)]:=factor(get(color))
122     a <- ggplot(tmp, aes_(x=as.name(regressor), y=quote(qxModel), color=as.name(color)))+
123       geom_line()+
124       facet_grid(paste0(vFacet, "~", hFacet), scales=Scales)+
125       labs(title=paste0("Attained Age: ", Age[1], ifelse(Age[2]>Age[1], paste0(" - ", Age[2]), "")),
126            subtitle=capt)
127     b <- ggplot(tmp, aes_(x=as.name(regressor), y=quote(ssaMultiple), color=as.name(color)))+
128       geom_line()+
129       facet_grid(paste0(vFacet, "~", hFacet), scales=Scales)+
130       labs(title=paste0("Attained Age: ", Age[1], ifelse(Age[2]>Age[1], paste0(" - ", Age[2]), "")),
131            subtitle=capt)
132   }
133   return(grid.arrange(a,b,nrow=2))
134 }
135 ...
136
137 # Model Predictions
138
139 Let's compare model predictions across different Issue Years
140
141 ***{r Duration Shape class }
142 # plotEffectsOneAge(regressor="duration", vFacet="Class", hFacet="Sex", color="IssueYear",
143 # IssueYear=c(1980, 1985, 1990, 1995, 2000, 2005), duration=c(1,30), trimult=FALSE, AttainedAge=40)
128:19 plotEffectsOneAge(regressor, vFacet, hFacet, Color, Scales, capt, ...)
```

Environment

Object	Size
DSOR	Large ds (137/649 elements, 12 MB)
bsdm	Large bs (1688535 elements, 12.9 mb)
dat	1088728 obs. of 34 variables
hmd	1870 obs. of 4 variables
JAMS_F	List of 10
JAMS_M	List of 10
smt	166 obs. of 3 variables
ssa	240 obs. of 3 variables

Values

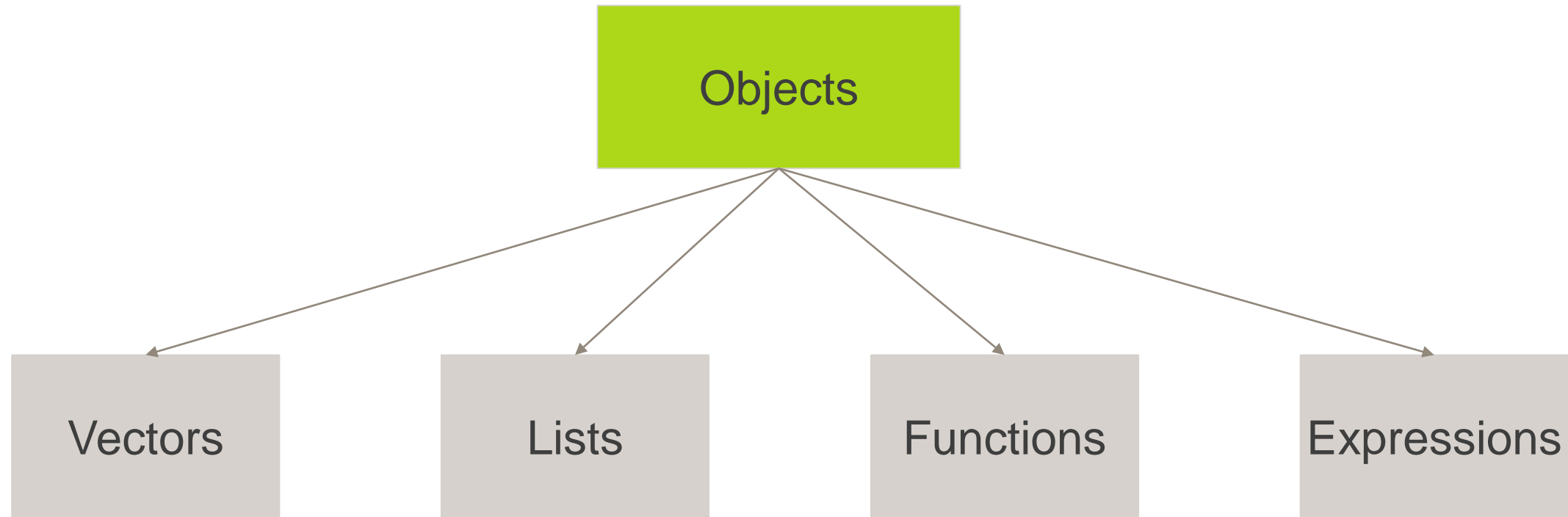
Form	"(bs (AttainedAge)+Class+bs(duration)+Fac..."
Gctorture	FALSE
root	"Q:/MRC_Reboot_GLM/Ultimates/Iteration3"

Functions

createDataSet	function (trimult = TRUE, AttainedAge ...
plotEffectsOne	function (regressor = "AvgFaceFactor", ...

Console

```
> class(createdDataSet())
[1] "function"
>
```

R Basic Structure – Object Classes

*somewhat
different*

Vectors

- ▶ numeric
- ▶ complex
- ▶ logical
- ▶ character
- ▶ factor
- ▶ raw

Lists

- ▶ list
- ▶ matrix
- ▶ array
- ▶ data.frame
- *data.table

Functions

- ▶ function

Expressions

- ▶ formulae
(in general, don't worry about expressions)

*somewhat
different*

Packages

hannover re[®]

What is an R Package?

- A collection of:
 - Functions (computer code)
 - In R or *another language*
 - Datasets
 - Help Files
 - Documentation
 - Possibly including “vignettes” (user guides)

```
install.packages("packageName")
```

```
library(packageName) -OR- require(packageName)
```

There are “Actuarial” Packages

- Example: actuar
 - These tend to be P&C-centric
 - May simply combine common statistical distribution functions
- survival very useful for Cox models!

**It's probably a bad idea to try to list every package
you might ever be interested in!**

Heavy Hitter Packages

- `data.table` – fast (written in C) handling of large data objects; some syntax sugar
- `tidyverse` – (more to come) Can make data wrangling user-friendly
- `ggplot2` – Robust graphics; Standard ‘grammar of graphics’
- `glmnet` – take your glm to the next level with regularization!

- `doParallel` – for advanced use, allows parallel processing

*somewhat
different*

Use Case



hannover re[®]

All code available on github!

*somewhat
different*

<https://github.com/hanewinckel/AnnualMeeting2019-R101>

pdf of rmarkown report also contains links to cheat sheets:

<https://rstudio.com/resources/cheatsheets/>

Mortality Analysis

Goals:

- Read Data (Review)
- Clean Data (Review)
- Analyze Data
 - (and show it to your boss!)
- Basic Modelling – Basis Adjustments

Twist – Data is so big it slows down R with dplyr!

Data

Publicly available ILEC data from 2009-2015 Experience Report:

<https://www.soa.org/resources/research-reports/2019/2009-2015-individual-life-mortality/>

For speed, we will use a subset (1MM) of rows:

```
pth <- 'C:/<File Path>/ILEC_2009-15 Data 20180601.txt'
```

```
dat <- fread(pth, stringsAsFactors = TRUE,  
            nrows=1000000, check.names = TRUE)
```

Read Speed

- To show the speed advantage of data.table over “base” R:

- Base R:

```
temp <- Sys.time()
invisible(read.csv(pth,header=TRUE,stringsAsFactors = TRUE,
                  nrows = 1000000))
print("read.csv time: "); print(Sys.time()-temp)
```

read.csv time: Time difference of 1.079535 mins

- data.table

```
temp <- Sys.time()
invisible(fread(pth,stringsAsFactors = TRUE,nrows = 1000000))
print("fread time: "); print(Sys.time()-temp)
```

fread time: Time difference of 2.733488 secs

*somewhat
different*

Data Wrangling

hannover re[®]

Rename Columns

- Not Trivial! Should be:
 - easy to remember/type
 - descriptive

```
setnames(dat, 'Number.of.Deaths', 'Deaths')
setnames(dat, 'Policies.Exposed', 'Exposure')
invisible(sapply(names(dat),
  function(x) setnames(dat, x, gsub('\\.', '', x))))
```

#Get names of fields

```
names(dat)
```

```
## [1] "ObservationYear" "CommonCompanyIndicator57"
## [3] "PreferredIndicator" "Gender"
## [5] "SmokerStatus" "InsurancePlan"
## [7] "IssueAge" "Duration"
## [9] "AttainedAge" "AgeBasis"
## [11] "FaceAmountBand" "IssueYear"
## [13] "NumberofPreferredClasses" "PreferredClass"
## [15] "SOAAnticipatedLevelTermPeriod" "SOAGuaranteedLevelTermPeriod"
## [17] "SOAPostleveltermindicator" "Select_Ultimate_Indicator"
## [19] "Deaths" "DeathClaimAmount"
## [21] "Exposure" "AmountExposed"
## [23] "ExpectedDeathQX7580EbyAmount" "ExpectedDeathQX2001VBTbyAmount"
## [25] "ExpectedDeathQX2008VBTbyAmount" "ExpectedDeathQX2008VBTLUbyAmount"
## [27] "ExpectedDeathQX2015VBTbyAmount" "ExpectedDeathQX7580EbyPolicy"
## [29] "ExpectedDeathQX2001VBTbyPolicy" "ExpectedDeathQX2008VBTbyPolicy"
## [31] "ExpectedDeathQX2008VBTLUbyPolicy" "ExpectedDeathQX2015VBTbyPolicy"
```

Remove/Clean Bad Data

#Show rows with no exposure

```
dat[Exposure==0, .N]
```

```
## [1] 6979
```

```
dat[AmountExposed==0, .N]
```

```
## [1] 6979
```

#Fix Rows with No Exposure

```
dat <- dat[Exposure > 0 & AmountExposed > 0]
```

Remove/Clean Bad Data

#Quantify non-level or unknown term products

```
table(dat$SOAGuaranteedLevelTermPeriod)
```

```
## 10 yr guaranteed 15 yr guaranteed 20 yr guaranteed 25 yr guaranteed
##           73134           47609           56041           1303
## 30 yr guaranteed 5 yr guaranteed  N/A (Not Term)  Not Level Term
##           23311           38725           633335           28938
##           Unknown
##           90625
```

- *#Remove non-level Term*
- `dat <- dat[!SOAGuaranteedLevelTermPeriod %in% c('Unknown', 'Not Level Term')]`
- *#Verify Removal*

```
table(dat$SOAGuaranteedLevelTermPeriod)
```

```
## 10 yr guaranteed 15 yr guaranteed 20 yr guaranteed 25 yr guaranteed
##           73134           47609           56041           1303
## 30 yr guaranteed 5 yr guaranteed  N/A (Not Term)  Not Level Term
##           23311           38725           633335           0
##           Unknown
##           0
```

*somewhat
different*

Plotting

hannover re[®]

ggplot2 – grammar of graphics

- Cheat Sheet: <https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf>

Format Data

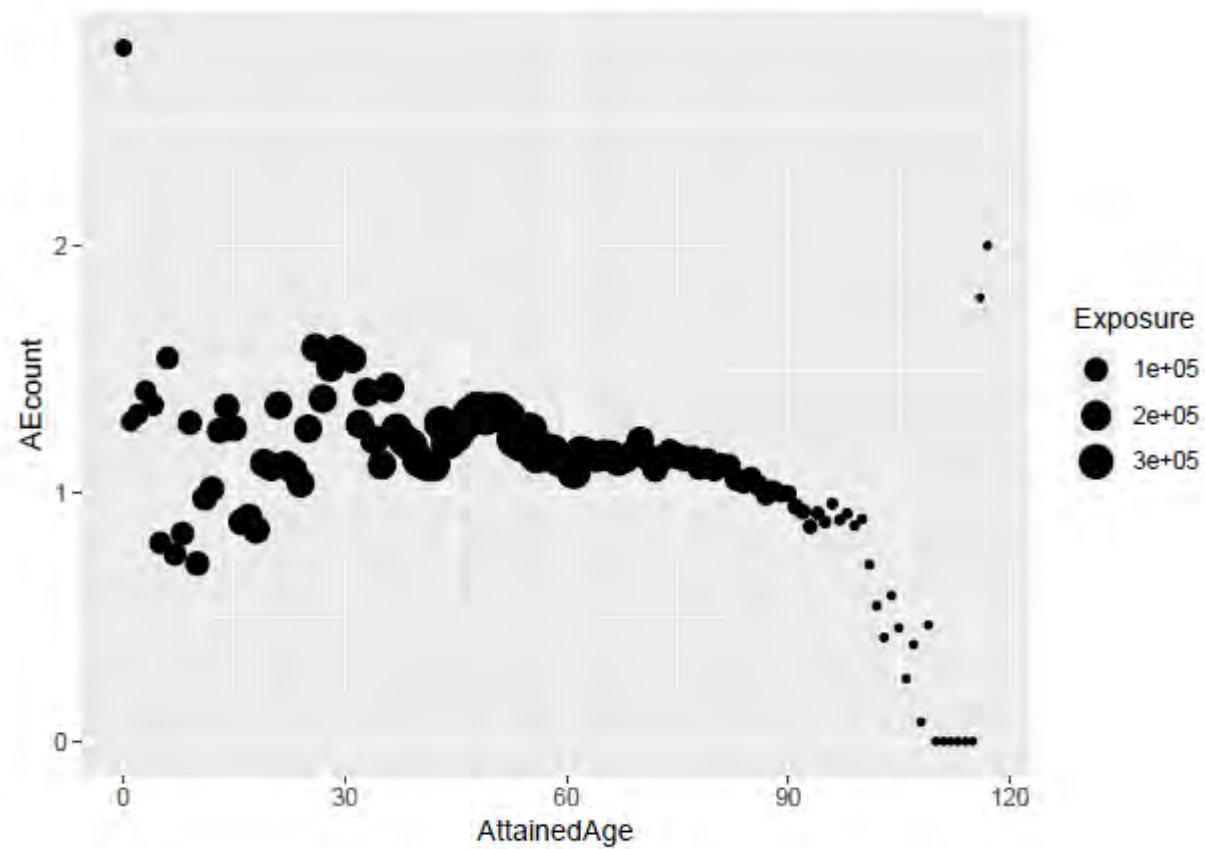
Data must be 'plottable' (formatted like your desired plot)

```
plotMeSimple <-  
dat[,.(Deaths=sum(Deaths), EDvbt15=sum(ExpectedDeathQX2015VBTbyPolicy),  
        AEcount=sum(Deaths)/sum(ExpectedDeathQX2015VBTbyPolicy),  
        Exposure=sum(Exposure)),  
      by=AttainedAge[order(AttainedAge)]  
#Note - I did not "need" [order(AttainedAge)], but it can be nice to have data  
ordered!
```

```
plotMeComplex <-  
dat[,.(Deaths=sum(Deaths), EDvbt15=sum(ExpectedDeathQX2015VBTbyPolicy),  
        AEcount=sum(Deaths)/sum(ExpectedDeathQX2015VBTbyPolicy),  
        Exposure=sum(Exposure)),  
      by=.(Duration, Gender, SmokerStatus, InsurancePlan)]
```

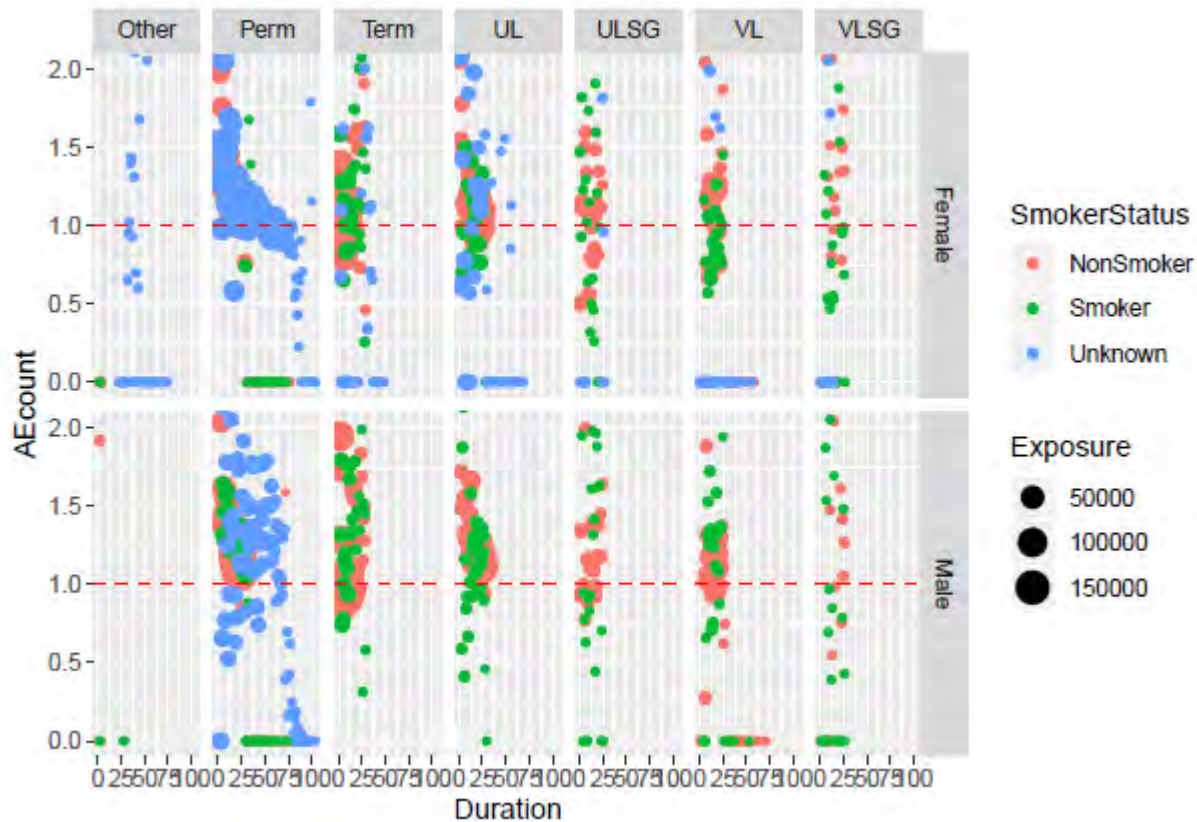
ggplot2 - Simple

```
ggplot(data=plotMeSimple, aes(x=AttainedAge, y=AEcount, size=Exposure)) +  
  geom_point()
```



ggplot2 - Complex

```
ggplot(data=plotMeComplex, aes(x=Duration, y=AEcount,
                               size=Exposure, color=SmokerStatus)) +
  geom_point() +
  facet_grid(Gender~InsurancePlan) +
  coord_cartesian(ylim=c(0, 2)) +
  geom_hline(yintercept=1, color='red', linetype='dashed')
```



*somewhat
different*

Modelling

hannover re[®]

Special Model Application!

- This is not a “modelling” 101
 - Still: this is what many users want to do
- So here’s a trick....
 - Make a given basis your “offset” term
 - Model will use your basis much like a “prior”
 - Model results tell how to “adjust” model!

$$\text{Offset} = \ln(\text{Exposure})$$

so for a glm model:

```
model <- glm(Deaths ~ (AttainedAge + Duration + Gender + SmokerStatus)^2,  
            data=dat, offset=log(ExpectedDeathQX2015VBTbyPolicy), family='poisson')
```

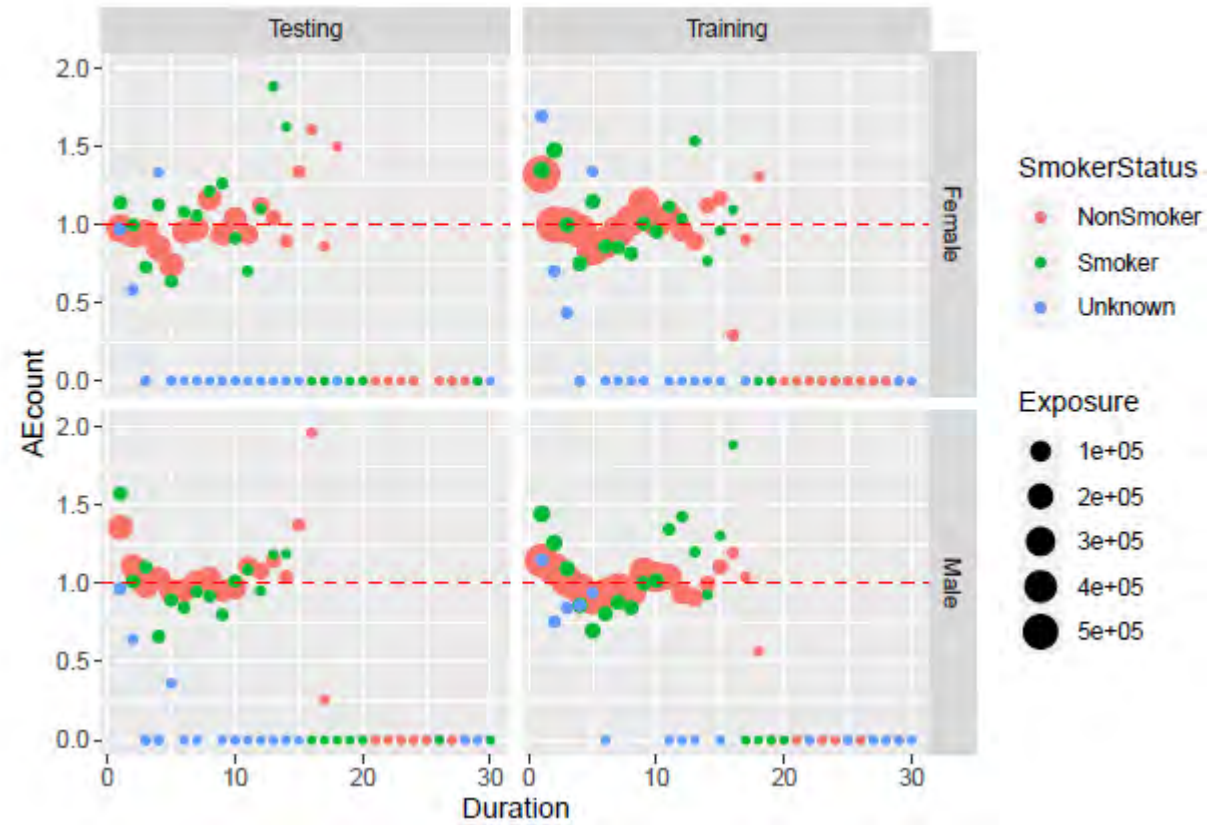
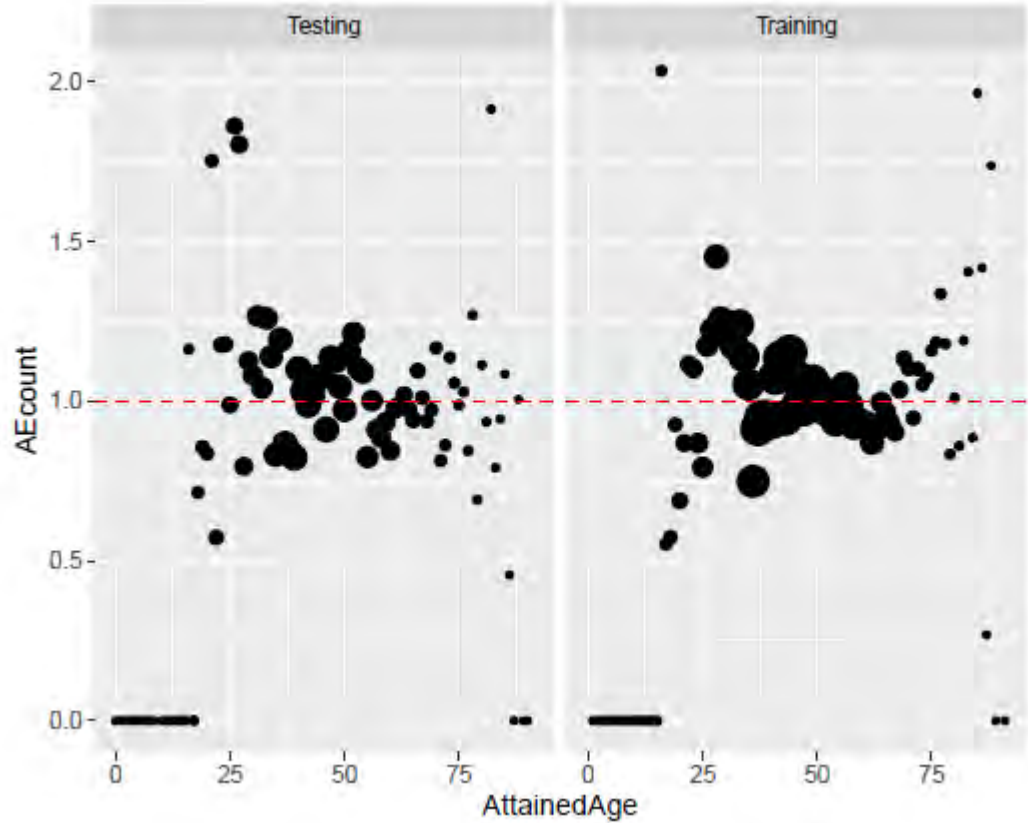
Testing/Training

- can use random numbers (runif())
- caret package used below

```
#Split Test/Train
trainingRows <- createDataPartition(dat$AttainedAge,p=.7,list=FALSE)
dat[,Set:='Testing']
dat[trainingRows,Set:='Training']

model <- glm(Deaths ~ (AttainedAge + Duration + Gender + SmokerStatus)^2,
             data=dat[Set=='Training'],
             offset=log(ExpectedDeathQX2015VBTbyPolicy),family='poisson')
```

Analyze Result



This is a simplified view of the modelling and analysis process.
This was to give an “R101”.

As your skills develop, you will add to your modelling and development toolkits!

hannover **re**[®]

SESSION 174 - R 101: AVAILABLE ACTUARIAL PACKAGES & CREATING REPRODUCIBLE ACTUARIAL ANALYSES

OCTOBER 30, 2019

Matthew Zhang, FSA, MAAA, CERA

CONFIDENTIALITY

Our clients' industries are extremely competitive, and the maintenance of confidentiality with respect to our clients' plans and data is critical. Oliver Wyman rigorously applies internal confidentiality practices to protect the confidentiality of all client information.

Similarly, our industry is very competitive. We view our approaches and insights as proprietary and therefore look to our clients to protect our interests in our proposals, presentations, methodologies and analytical techniques. Under no circumstances should this material be shared with any third party without the prior written consent of Oliver Wyman.

© Oliver Wyman

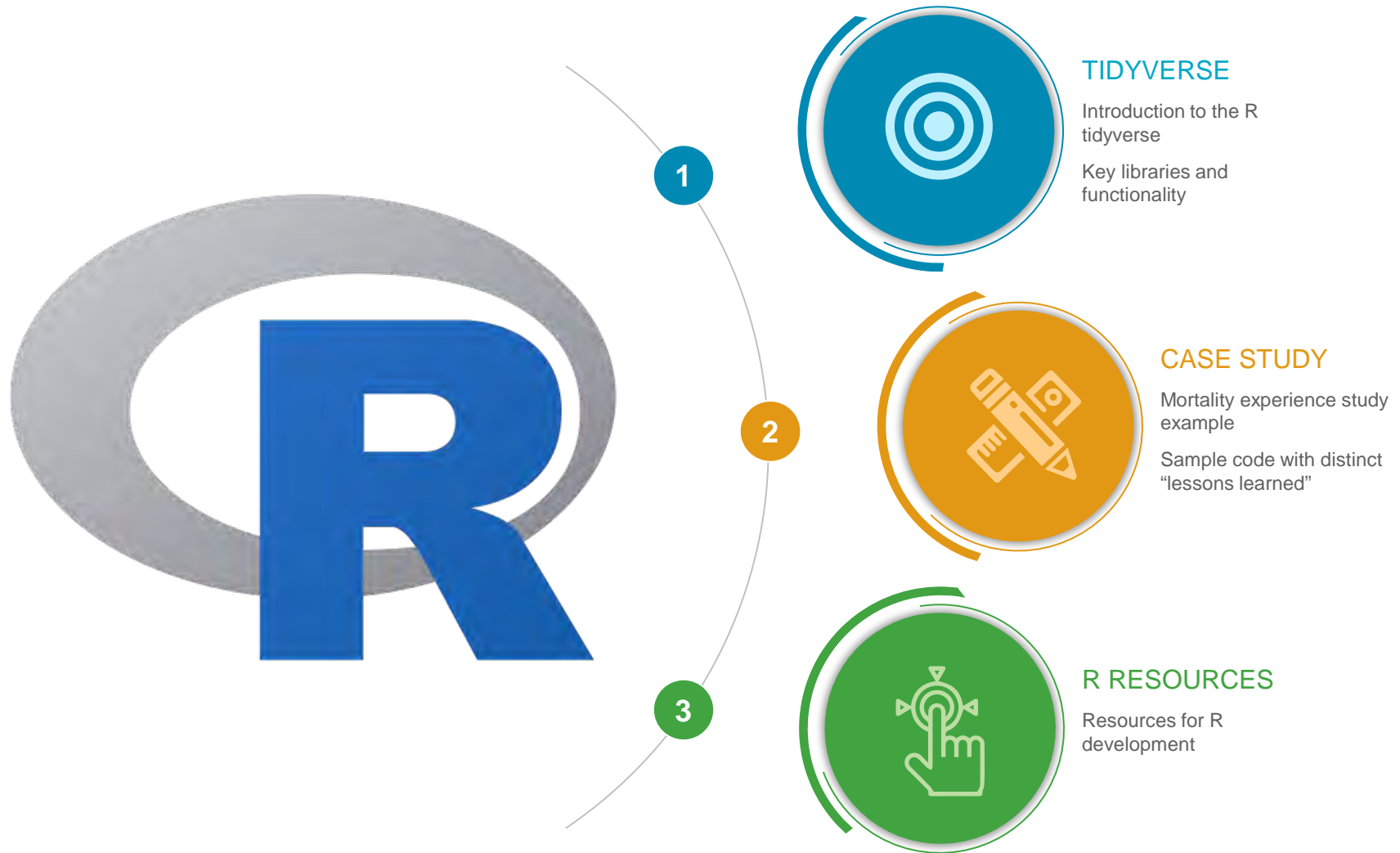
Contents

1. Introduction	4
2. Tidyverse	6
3. Case study	10
4. R resources	20

1 | Introduction

Introduction and objectives

A deep-dive into common R concepts and specific code samples will provide you an expanded toolset to deploy in your day-to-day work



2 | Tidyverse

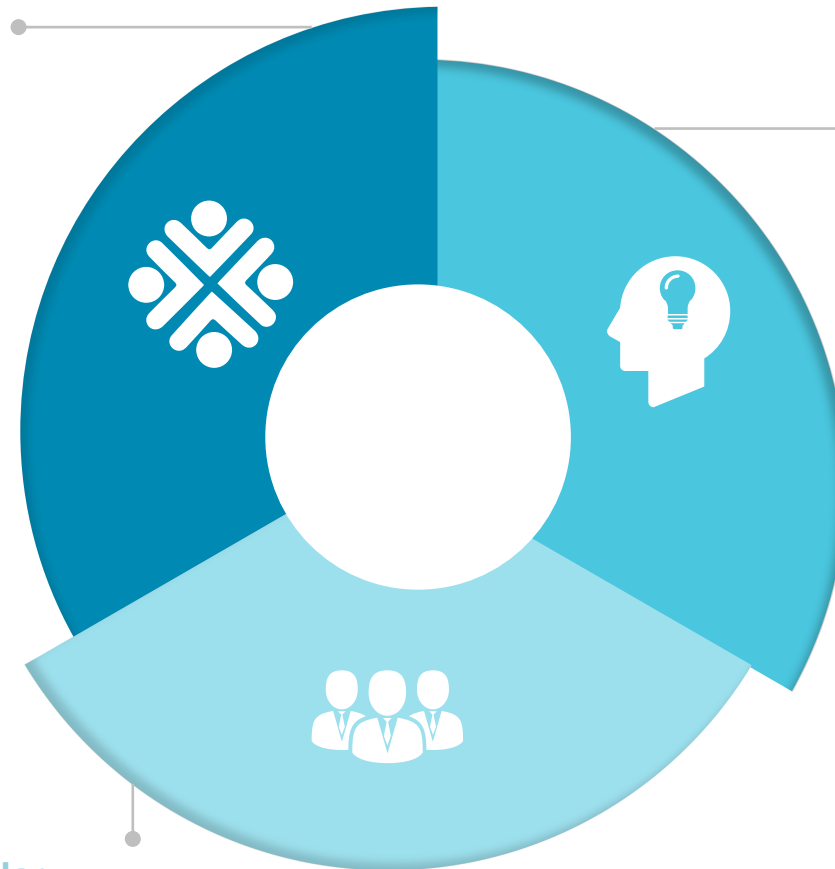
Introduction to the tidyverse

The examples in this presentation center around the tidyverse suite of R packages

Powerful integrated workflow

Common worksteps within typical data analysis exercises are well supported:

- Importing data
- Tidying data
- Transforming data
- Modeling
- Presentation and visualization



Consistent design

Across multiple dimensions, the tidyverse is designed to be internally consistent:

- Use of 'tidy' data
- Pipes
- Uniformity in structures 'under the hood'

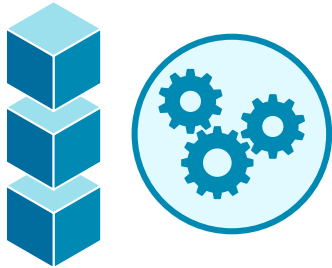
Popular

The tidyverse is both very popular and well supported, granting access to a wealth of training resources and a wide support network

Key libraries

Three tidyverse libraries underpin the examples in this presentation

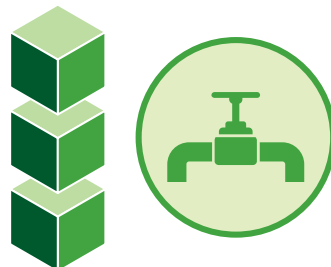
Data manipulation *dplyr*



- Key verbs (select, filter, mutate, group) enable a robust data wrangling toolkit
- Benefits significantly from the use of pipes as a part of the tidyverse philosophy
- Makes significant use of 'tibble' data structures, an alternative to traditional data.frames

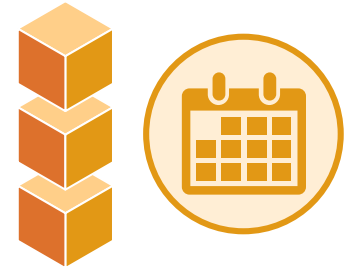


Pipes *magrittr*



- Increases the readability of code into natural processes
- Analogy to Power Query / Microsoft BI
- Key tidyverse packages are designed to work well with pipes
- Cuts down the number of intermediate variables

Time and date *lubridate*



- Powerful tools are available to parse, set, and perform often frustrating date and time calculations with ease
- Complex considerations of leap days, daylight savings, time zones, and other quirks are handed out-of-the-box

Code example: Piping

Sample code

```
library(tidyverse) 1
# pipes
iris %>%
  group_by(Species) %>%
  summarize_if(is.numeric, mean) %>%
  ungroup() %>%
  gather(measure, value, -Species) %>%
  arrange(value) 2

# alternative 1 - nesting
arrange(gather(ungroup(summarize_if(group_by(iris, Species),
  is.numeric, mean)), measure, value, -Species))

# alternative 2 - intermediate steps 3
step1 <- iris
step2 <- group_by(step1, Species)
step3 <- summarize_if(step2, is.numeric, mean)
step4 <- ungroup(step3)
step5 <- gather(step4, measure, value, -Species)
step6 <- arrange(step5)
```

Key functionality

1. The magrittr library, which is bundled with tidyverse packages, enables pipe operators

Simple pipes:

$x \%>\% f$ is equivalent to $f(x)$

$x \%>\% f(y)$ is equivalent to $f(x, y)$

$x \%>\% f(y, .)$ is equivalent to $f(y, x)$

2. Compatibility with pipe-based workflows is a basic principle in the tidyverse and functions in key libraries such as dplyr and ggplot2 work seamlessly with pipes
3. Pipes offer significant advantages over alternative workflows in terms of code readability and access

2 | Case study

Overview of the case

Each step of the mortality experience study process has distinct challenges for R to resolve



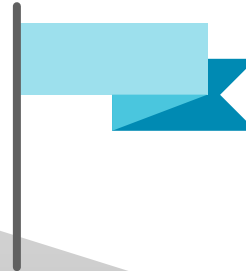
Present results

- Assess results by flexible cohorts



Calculate exposures and mortality

- Exposure calculations require significant processing, including date and time calculations
- Mortality rates must be properly derived
- Large volumes of calculations need to be performed



Process inforce data

- Data from a variety of formats need to be “cleaned”
- The process should be repeatable and self documenting



Processing inforce data



PROBLEM

- Data comes from a variety of sources (databases, text, csv, etc.) in formats which may not be consistent
- The processing of dates can be cumbersome and error prone



SOLUTION

- R has a robust toolkit of flexible and parameterizable functions for the importing of data from all sources
- Powerful date and time processing functions are available out-of-the-box
- R allows the raw data, processing mechanism, and post-processed outcomes to be separated to support fully auditable and repeatable workflows
- Data can be easily transformed into data structures which support processing and storage

Example code: Importing data



Sample code

```
## Importing key libraries
library(tidyverse)
library(lubridate)

# reading Excel data
excel_data_column_remaps <- c('ID', 'RISK_CLASS', 'GENDER', 'ISSUE_DATE',
' BIRTH_DATE', 'FACE', 'DEATH_DATE', 'EXP_YEAR', 'SMOKER')
data_2010_2011 <- readxl::read_xlsx('raw data/Legacy format policy data.xlsx',
range = '2010-2011!A5:I19421', col_names = excel_data_column_remaps)
data_2012_2013 <- readxl::read_xlsx('raw data/Legacy format policy data.xlsx',
sheet = '2012-2013', skip = 4, col_names = excel_data_column_remaps)

excel_data <- bind_rows(data_2010_2011, data_2012_2013)

# reading flat data
csv_data <- read_csv(file = 'raw data/2014_plus.csv')
```

Sample input

2014_plus.csv - Notepad

```
File Edit Format View Help
ID,SMOKER,RISK_CLASS,GENDER,ISSUE_DATE,BIRTH_DATE,FACE,DEATH_DATE,EXP_YEAR
DTL1000018,N,Std_NS,M,19980925,19501209,100000,0,2014
DTL1000018,N,Std_NS,M,19980925,19501209,100000,0,2015
DTL1000018,N,Std_NS,M,19980925,19501209,100000,0,2016
DTL1000054,N,Std_NS,M,20090528,19571025,1300000,0,2014
DTL1000054,N,Std_NS,M,20090528,19571025,1300000,0,2015
DTL1000054,N,Std_NS,M,20090528,19571025,1300000,0,2016
DTL1000058,S,Pref_SM,F,20021230,19380127,80000,0,2014
```

Administration systems output: 2010-2011

Policy ID	RISK_CLASS	GENDER	ISSUE_DATE	BIRTH_DATE	FACE	DEATH_DATE	EXP_YEAR	Smoker status
DTL1000018	Std_NS	Male	25-09-1998	08-12-1950	100000	NA	2010	Non-Smoker
DTL1000018	Std_NS	Male	25-09-1998	08-12-1950	100000	NA	2011	Non-Smoker
DTL1000054	Std_NS	Female	28-05-2009	25-10-1957	1300000	NA	2010	Non-Smoker
DTL1000054	Std_NS	Female	28-05-2009	25-10-1957	1300000	NA	2011	Non-Smoker
DTL1000058	Pref_SM	Female	30-12-2002	27-01-1938	80000	NA	2010	Smoker
DTL1000058	Pref_SM	Female	30-12-2002	27-01-1938	80000	NA	2011	Smoker
DTL1000062	Pref_NS	Male	01-09-1996	27-12-1946	1800000	NA	2010	Non-Smoker
DTL1000062	Pref_NS	Male	01-09-1996	27-12-1946	1800000	NA	2011	Non-Smoker
DTL1000074	Pref_Elite_NS	Male	08-10-1999	17-03-1973	240000	NA	2010	Non-Smoker
DTL1000074	Pref_Elite_NS	Male	08-10-1999	17-03-1973	240000	NA	2011	Non-Smoker
DTL1000112	Pref_Elite_NS	Female	27-12-1999	19-12-1962	180000	NA	2010	Non-Smoker

Key functionality

1. Installed R libraries can be loaded into sessions for easy access, but functions in libraries can also be called directly without loading
2. R has a wealth of read functions from various libraries; these functions generally have powerful parameters to help interpret data from different formats
3. The `dply::bind_*` functions are used to combine different sets of data

Sample output

ID	SMOKER	RISK_CLASS	GENDER	ISSUE_DATE	BIRTH_DATE	FACE	DEATH_DATE	EXP_YEAR
DTL1000018	N	Std_NS	M	19980925	19501209	100000	0	2014
DTL1000018	N	Std_NS	M	19980925	19501209	100000	0	2015
DTL1000018	N	Std_NS	M	19980925	19501209	100000	0	2016
DTL1000054	N	Std_NS	F	20090528	19571025	1300000	0	2014
DTL1000054	N	Std_NS	F	20090528	19571025	1300000	0	2015
DTL1000054	N	Std_NS	F	20090528	19571025	1300000	0	2016
DTL1000058	S	Pref_SM	F	20021230	19380127	80000	0	2014

Example code: Cleaning, consolidating, and saving data



Sample code

```
# cleaning data
excel_data_clean <- excel_data %>%
  mutate(
    1 ISSUE_DATE = dmy(ISSUE_DATE),
    DEATH_DATE = dmy(DEATH_DATE),
    BIRTH_DATE = dmy(BIRTH_DATE),
    SMOKER = substr(SMOKER, 1, 1),
    GENDER = substr(GENDER, 1, 1)
  )
  2

csv_data_clean <- csv_data %>%
  mutate(
    ISSUE_DATE = ymd(ISSUE_DATE),
    DEATH_DATE = ymd(DEATH_DATE),
    BIRTH_DATE = ymd(BIRTH_DATE)
  )
  2


clean_data <- bind_rows(excel_data_clean, csv_data_clean)

# writing clean data
write_csv(clean_data, path = 'data/clean_data.csv')
  3
```

Key functionality

1. The **dply::mutate** function is one of the most important 'verbs' and is used to either **modify** or **create** new variables
2. The lubridate package allows dates to be parsed intelligently from a variety of formats, enabling powerful manipulation and measurement functions
3. Data in R can be easily output in a variety of formats for use by external processes or to leave an audit trail

Output

 clean_data.csv

ID	RISK_CLASS	GENDER	ISSUE_DATE	BIRTH_DATE	FACE	DEATH_DATE	EXP_YEAR	SMOKER
DTL1000018	Std_NS	M	1998-09-25	1950-12-09	100000	NA	2010	N
DTL1000018	Std_NS	M	1998-09-25	1950-12-09	100000	NA	2011	N
DTL1000054	Std_NS	F	2009-05-28	1957-10-25	1300000	NA	2010	N
DTL1000054	Std_NS	F	2009-05-28	1957-10-25	1300000	NA	2011	N
DTL1000058	Pref_SM	F	2002-12-30	1938-01-27	80000	NA	2010	S
DTL1000058	Pref_SM	F	2002-12-30	1938-01-27	80000	NA	2011	S
DTL1000062	Pref_NS	M	1996-09-01	1946-12-27	1800000	NA	2010	N

Calculate exposures and mortality



PROBLEM

- A large volume of inforce data will need to be processed to calculate exposure accurately
- The data will need to be validated to check for faulty data and other limitations, such as the study window
- For each exposure record, expected mortality must be calculated
- Mortality tables will need to be selected and processed



SOLUTION

- No time is wasted in manually managing rows or shape of data
- All manipulations are flexible and can be extended naturally
- No limitation on the volume of data
- Powerful tools exist to transform data into a tidy format
- Faster processing compared to common numerical tools, such as Excel

Example code: Calculating actual mortality exposures



Sample code

```
# splitting data across multiple rows
data_study <- data %>%
  slice(rep(1:n(), each = 3)) 1

# filtering policies and calculating exposures
data_study <- data_study %>%
  mutate(
    PERIOD_TYPE = rep(c('a', 'b', 'c'), nrow(data)),
    DATE1 = `year<-` (ISSUE_DATE, EXP_YEAR),
    DATE2 = `year<-` (BIRTH_DATE, EXP_YEAR),
    PERIOD_START = if_else(PERIOD_TYPE == 'a',
                          ymd(paste0(EXP_YEAR, '0101')),
                          if_else(PERIOD_TYPE == 'b',
                                    pmin(DATE1, DATE2),
                                    pmax(DATE1, DATE2))),
    PERIOD_END = if_else(PERIOD_TYPE == 'c',
                        ymd(paste0(EXP_YEAR, '1231')),
                        if_else(PERIOD_TYPE == 'b',
                                  pmax(DATE1, DATE2),
                                  pmin(DATE1, DATE2))),
    ATTAINED_AGE_LAST = floor(interval(BIRTH_DATE, PERIOD_START) / years(1)
  ),
  DURATION = floor(interval(ISSUE_DATE, PERIOD_START) / years(1) + 1) %>%
  filter(
    PERIOD_START < if_else(is.na(DEATH_DATE), ymd(20991231), DEATH_DATE),
    PERIOD_START >= ISSUE_DATE,
    PERIOD_START >= ymd(20130101), # start of study period
    PERIOD_END <= ymd(20201231) # end of study period
  ) %>%
  mutate(
    DEATH_IND = if_else(is.na(DEATH_DATE),
                       FALSE,
                       if_else(DEATH_DATE >= PERIOD_START & DEATH_DATE <=
    PERIOD_END,
                              TRUE,
                              FALSE)),
    EXPOSURE_POL = interval(PERIOD_START, PERIOD_END) / years(1),
    EXPOSURE_FACE = EXPOSURE_POL * FACE,
    ACTUAL_COUNT = EXPOSURE_POL * as.numeric(DEATH_IND),
    ACTUAL_FACE = EXPOSURE_FACE * as.numeric(DEATH_IND)
  )
```

Key functionality

1. The **dplyr::slice** “verb” is used to select rows by position, and is used here to create repeated rows
2. Through the use of pipes and the native ability of dplyr verbs to stack parameters, a relatively complex procedural set of data manipulations can be presented and executed cleanly
3. The **dplyr::filter** verb chooses rows based on conditions, and is often an essential step in data wangling exercises – the use of pipes allows data filters to be easily embedded in workflows
4. The **lubridate::interval** function is one of many functions available to process date and time data – simple syntax masks powerful functionality, such as awareness of leap years, support for time zones, and accessible date arithmetic functions

Example code: Table reading and expected mortality



Sample code

```
# reading and converting mortality tables
mort_tables <- read_csv('data/mortality_tables.csv') %>%
gather("ISSUE_AGE", "MORTALITY", IA1:IA100) %>%
mutate(ISSUE_AGE = as.integer((gsub("\\D", "", ISSUE_AGE))))

# calculating expected mortality
data_study_combined <- data_study %>%
mutate(ISSUE_AGE = floor(interval(BIRTH_DATE, ISSUE_DATE) / years(1)),
LOOKUP_DUR = pmin(DURATION, 26),
LOOKUP_TABLE_NAME = paste('2008 VBT', GENDER, SMOKER, 'L', sep = ' ')) %>%
left_join(mort_tables, c('LOOKUP_DUR' = 'DUR', 'LOOKUP_TABLE_NAME' = 'TABLE_NAME',
'ISSUE_AGE' = 'ISSUE_AGE')) %>%
mutate(EXP_COUNT = EXPOSURE_POL * MORTALITY / 1000,
EXP_FACE = EXPOSURE_FACE * MORTALITY / 1000)
```

Key functionality

1. The **tidyr::gather** function is a key verb used to transform data into the 'tidy' format used across the tidyverse framework with ease, simplifying the data analysis process
2. The **gsub** function is a part of base R's pattern matching and replacement functionality which can leverage regular expressions to perform powerful and concise manipulations – this flexible mechanic is lightly used here to extract numerical information from strings
3. The **dplyr::left_join** function is one among a suite of generic functions to combine data sets

Output

ID	RISK_CLASS	GENDER	ISSUE_DATE	PERIOD_START	PERIOD_END	DEATH_IND	EXPOSURE_POL	EXPOSURE_FACE
DTL1000018	Std_NS	M	1998-09-25	2013-01-01	2013-09-25	FALSE	0.731506849	73150.6849
DTL1000018	Std_NS	M	1998-09-25	2013-09-25	2013-12-09	FALSE	0.205479452	20547.9452
DTL1000018	Std_NS	M	1998-09-25	2013-12-09	2013-12-31	FALSE	0.060273973	6027.3973
DTL1000054	Std_NS	F	2009-05-28	2013-01-01	2013-05-28	FALSE	0.402739726	523561.6438

Present results



PROBLEM

- Analysis requires periodic reproduction of summaries, graphs, and other forms of presentation – often a time consuming and repetitive process
- Limited visualization solutions for exotic graphics exist within common tools such as Excel
- Documentation of sources of results and numbers may be imperfect or overlooked during time crunch; process may not be easily repeatable or auditable



SOLUTION

- Summaries and graphs can be easily reproduced with refreshed data
- The ggplot2 package from the tidyverse is an extremely versatile and robust framework for visualization and graphing
- R code is self documenting and all outputs are reproducible

Example code: Grouping and summarizing



Sample code

```
# grouping and summarizing data
tbl_gender <- data_study_combined %>%
  group_by(GENDER) ①

tbl_gender %>%
  summarise(② ACTUAL_COUNT = sum(DEATH_IND),
            ACTUAL_FACE = sum(ACTUAL_FACE),
            EXP_COUNT = sum(EXP_COUNT),
            EXP_FACE = sum(EXP_FACE),
            A_E_POL = ACTUAL_COUNT/EXP_COUNT,
            A_E_FACE = ACTUAL_FACE/EXP_FACE)
```

Key functionality

1. The dplyr library uses the 'tibble' data structure that inherently contains support for grouping levels
2. The `dplyr::summarise` data allows a more powerful and flexible deployment of aggregate data – think dynamic Excel Pivot Tables

Output





```
> tbl_gender
# A tibble: 110,950 x 27
# Groups:   GENDER [2]
  ID RISK_CLASS GENDER ISSUE_DATE BIRTH_DATE FACE DEATH_DATE
  <chr> <chr> <chr> <date> <date> <dbl> <date>
1 DTL1~ Std_NS M 1998-09-25 1950-12-09 1.00e5 NA
2 DTL1~ Std_NS M 1998-09-25 1950-12-09 1.00e5 NA
3 DTL1~ Std_NS M 1998-09-25 1950-12-09 1.00e5 NA
4 DTL1~ Std_NS F 2009-05-28 1957-10-25 1.30e6 NA
5 DTL1~ Std_NS F 2009-05-28 1957-10-25 1.30e6 NA
6 DTL1~ Std_NS F 2009-05-28 1957-10-25 1.30e6 NA
7 DTL1~ Pref_SM F 2002-12-30 1938-01-27 8.00e4 NA
```

GENDER	ACTUAL_COUNT	ACTUAL_FACE	EXP_COUNT	EXP_FACE	A_E_POL	A_E_FACE
F	278	42908562	120.6136	41209976	2.30488	1.0412178
M	406	66268235	249.4638	103787358	1.62749	0.6385001

4 | R resources

Publicly available resources

A wealth of educational resources is available as R is both popular and well established in many fields

 <p>OFFICIAL SOURCES</p>	<ul style="list-style-type: none">• Built-in training packages (Swirl)• CRAN• R cheat sheet for tidyverse packages (dplyr, lubridate, etc.)• Free “R for Data Science” ebook https://r4ds.had.co.nz/• “?” documentation	<p>Scan me!</p> 
 <p>THIRD-PARTY SOURCES</p>	<ul style="list-style-type: none">• Online training, both free and fee-based (YouTube, LinkedIn Learning, Udemy, DataCamp, Coursera, Lynda, etc.)• Numerous certification courses available	
 <p>COMMUNITY</p>	<ul style="list-style-type: none">• “Google it”• Stack Exchange• Kasa AI• “Ask around the office”	

QUALIFICATIONS, ASSUMPTIONS AND LIMITING CONDITIONS

This report is for the exclusive use of the Oliver Wyman client named herein. This report is not intended for general circulation or publication, nor is it to be reproduced, quoted or distributed for any purpose without the prior written permission of Oliver Wyman. There are no third party beneficiaries with respect to this report, and Oliver Wyman does not accept any liability to any third party.

Information furnished by others, upon which all or portions of this report are based, is believed to be reliable but has not been independently verified, unless otherwise expressly indicated. Public information and industry and statistical data are from sources we deem to be reliable; however, we make no representation as to the accuracy or completeness of such information. The findings contained in this report may contain predictions based on current data and historical trends. Any such predictions are subject to inherent risks and uncertainties. Oliver Wyman accepts no responsibility for actual results or future events.

The opinions expressed in this report are valid only for the purpose stated herein and as of the date of this report. No obligation is assumed to revise this report to reflect changes, events or conditions, which occur subsequent to the date hereof.

All decisions in connection with the implementation or use of advice or recommendations contained in this report are the sole responsibility of the client. This report does not represent investment advice nor does it provide an opinion regarding the fairness of any transaction to any and all parties.

Q&A