# Should Actuaries Use Exact Exposures?

Philip Adams

## Yes

…but, you will need to decide that for yourself. Below, I lay out my case for why life insurance experience studies should use only exact exposures. In short,

- Exact exposures are natural for risk measurement. The exposure is identical to when the company is on the risk. No more, no less.
- Predictive analytics and machine learning are only as sound as the data that go into them. Exact exposures ensure that there is nothing biasing the math.
- Other methods are at best approximations to exact exposures and bring with them their own risks. If an actuary chooses to go this route anyway, they must work around these issues.

The examples below cover situations I have frequently encountered when doing experience studies.

## The Problem

When someone dies, stops payment of premium or fully surrenders their policy, we run into challenges in measuring the rate or probability of that event. Ideally, we could design an experience study which gives us whatever we want, whenever we want.

An "ideal" study should include the following features:

- We can slice and dice the exposures and events along any dimension, including time and at desired granularity. Time dimensions would include whatever is relevant, such as time since issue and calendar time.
- The study makes monitoring easy, where we can track emerging experience against expectations accurately. The baselines and measurements should be consistent with whatever our needs are. For example, if I need to project mortality, I should be able to readily adapt the information in the study to whatever model I am using.

- The study should not get in the way of AI/ML/analytics. That is, I can take data from the study and use modern modeling methods.

Reality, though, is not so kind.

There are several fundamental and interrelated issues about calculating exposure that can get in the way of these objectives.

- Measuring exposure exactly seems very natural. We are on the risk from issue until the policy terminates, such as due to death, lapse, or surrender. However, the ratio of deaths to exposures is not a mortality rate. In fact, the formulas to convert this to a probability yield non-traditional answers to mortality estimates.
- One method to pad the exposure is to add half the deaths to the exposure. The ratio of deaths to the adjusted exposure will yield an approximate probability for cells with enough data. But the one-half is an approximation that only works where there is enough data, where the granularity of the study is annual, and where the uniform distribution of deaths assumption is valid. It is still possible in smaller cells for the ratio to be greater than 1. Thus the resulting ratio is close to, though not always close to, the true probability.
- Another to method to pad the exposure is to extend the exposure for the entire policy year. For example, if someone died on January 2, and their next policy anniversary is April 1, then the missing exposure from January 3 to March 31 is added to the exposure listing. The downside is that some fraction of the exposures are getting pushed to a future calendar period. This becomes an issue when viewing experience by calendar year, as the dangling exposure record depresses future ratios by inflating future exposures.

All of these items can cause problems with predictive modeling. In the first case, I can still use Poisson or Cox models, but the resulting models do not yield probabilities (still). In the second case, there could be bias in small cells, although I am now dealing with probabilities. In the third case, the model will reflect the exposure issues, especially for larger probabilities, if calendar year is a variable in the model.

Unfortunately, there is no free lunch, and it may be necessary to apply different solutions as needed.

We will walk though some examples to show how things can go wrong with each of these methods.

## Setting

If you look at the 2015 VBT table, you will see something curious at the end. The mortality rate levels off to 0.5 starting at age 112. We are going to exploit this extreme to illustrate how these various methods can break down. We will also attach the expected mortality rate to see how accurate each method is at getting to the true rate.

Suppose we have a portfolio of 100,000 lives issued 112+. Since the assumed mortality rate does not matter here, the exact age doesn't either. We have no idea where we picked up this legion of the superannuated.

To make matters a little more complicated, we add a second, non-claim termination with annual rate of 25%. The only thing this does is to censor exposures.

## Setting Up The Census

```
library(data.table)
library(tidyverse)
library(flextable)

nHorizon <- 3 # number of years of the study
nSims <- 100000 # number of lives

# First decrement - mortality
annual_rate <- 0.5
daily_rate <- 1-(1-annual_rate)^(1/365)

# Second decrement - lapse
annual_rate2 <- 0.25
modal_rate2 <- 1-(1-annual_rate2)^(1/12)

set.seed(0xBEEF)

# Caching
bUseCache <- TRUE
bInvalidateCache <- FALSE
cacheRoot <- file.path(".","caches")

# Function to convert a vector of life table mortality rates into a vector
# suitable for sampling
source("lifetable_to_distribution.R")

# Simulate the death days
death_table <- lifetable_to_distribution(rep(daily_rate,365*nHorizon))
events <- death_table[,sample(x=time,size=nSims,replace=T,prob=q_xpt)]

# Simulate the lapse days
lapse_table <- lifetable_to_distribution(rep(modal_rate2,nHorizon*12))
```

```
events2 <- lapse_table[,sample(x=time,size=nSims,replace=T,prob=q_xpt)]

# Set up the start and end dates of the study
study_start <- as_datetime(ymd(20210101))
study_end <- study_start %m+% years(nHorizon) %m+% days(-1)

# Set the granularity in months of the time dimensions
pol_period_granularity <- 12 # months per policy period
cal_period_granularity <- 12 # months per calendar period

cal_yr_breaks <- (study_start %m+% days(-1)) %m+%
  months(
    cal_period_granularity*(1:(interval(study_start %m+% days(-1),study_end) %/%
                                 months(cal_period_granularity)))
  )

cacheFile <- file.path(
       cacheRoot,
       "measuring-risk-census.parquet"
     )

if(bUseCache &
   !bInvalidateCache &
   file.exists(
     cacheFile
   )) {
  census <- arrow::read_parquet(
    cacheFile
  )
} else {
  # Create the census of policies, including simulated dates
  data.table(
    PolID=1:100000,
    Issue_Date=study_start %m+% days(sample.int(365,100000,replace=T)),
    Prem_Mode_Months=12
  ) %>%
    mutate(  Death_Date=Issue_Date %m+% days(events),
             Lapse_Date=Issue_Date %m+% months(events2)) ->
    census

  # Set the Termination Date, which is the earlier of the Death Date and
  # Lapse Date, unless the dates are after the study.
```

```
  census %>%
    mutate(Term_Date = as_datetime(
      ifelse(Death_Date <= Lapse_Date & Death_Date <= study_end,
             Death_Date,
             NA))) %>%
    mutate(Term_Date = as_datetime(
      ifelse(Lapse_Date <= Death_Date & Lapse_Date <= study_end,
             Lapse_Date,
             Term_Date))) ->
    census

  if(bUseCache)
    arrow::write_parquet(
      x=census,
      sink=cacheFile
    )
}


census %>%
  head() %>%
  flextable() %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| PolID | Issue_Date | Prem_Mode_Months | Death_Date | Lapse_Date | Term_Date |
|---|---|---|---|---|---|
| 1 | 2021-09-06 00:00:00 | 12 | 2022-04-15 00:00:00 | 2024-10-06 00:00:00 | 2022-04-15 00:00:00 |
| 2 | 2021-02-06 00:00:00 | 12 | 2021-09-18 00:00:00 | 2023-01-06 00:00:00 | 2021-09-18 00:00:00 |
| 3 | 2021-05-24 00:00:00 | 12 | 2023-01-21 00:00:00 | 2024-06-24 00:00:00 | 2023-01-21 00:00:00 |
| 4 | 2021-10-05 00:00:00 | 12 | 2023-05-05 00:00:00 | 2023-02-05 00:00:00 | 2023-02-05 00:00:00 |
| 5 | 2021-12-03 00:00:00 | 12 | 2022-04-16 00:00:00 | 2025-01-03 00:00:00 | 2022-04-16 00:00:00 |
| 6 | 2021-09-27 00:00:00 | 12 | 2024-09-27 00:00:00 | 2024-10-27 00:00:00 | |

## Expanding the Exposures

We then expand the census into a series of exposures.

```
source('expand.exposures.R')

cacheFile <- file.path(
  cacheRoot,
  "measuring-risk-exposures.parquet"
)

if(bUseCache &
   !bInvalidateCache &
   file.exists(
     cacheFile
   )
) {
  exposures <- arrow::read_parquet(
    cacheFile
  )
} else {
  census %>%
    filter(Issue_Date <= study_end) %>%
    select(PolID,Issue_Date,Term_Date,Prem_Mode_Months) %>%
    expand_exposures(
      .exp_period_start = study_start,
      .exp_period_end = study_end,
      .cal_yr_breaks = cal_yr_breaks,
      .pol_period_granularity = pol_period_granularity,
      .cal_period_granularity = cal_period_granularity,
      .issue_date = Issue_Date,
      .term_date = Term_Date,
      .ID=PolID,
      .prem_mode_months = Prem_Mode_Months,
      .exact_terminations = FALSE
    ) ->
    exposures

  if(bUseCache) {
    arrow::write_parquet(
      x=exposures,
      sink=cacheFile
    )
  }
}
```

```
exposures %>%
  head() %>%
  flextable() %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| PolID | monthaversary | exp_period_start | exp_period_end | pol_duration | exposure |
|---|---|---|---|---|---|
| 1 | 3.806452 | 2021-09-06 00:00:00 | 2021-12-31 00:00:00 | 1 | 0.3205479 |
| 1 | 7.300000 | 2022-01-01 00:00:00 | 2022-04-15 00:00:00 | 1 | 0.2876712 |
| 2 | 7.400000 | 2021-02-06 00:00:00 | 2021-09-18 00:00:00 | 1 | 0.6164384 |
| 3 | 7.225806 | 2021-05-24 00:00:00 | 2021-12-31 00:00:00 | 1 | 0.6082192 |
| 3 | 12.000000 | 2022-01-01 00:00:00 | 2022-05-23 00:00:00 | 1 | 0.3917808 |
| 3 | 19.225806 | 2022-05-24 00:00:00 | 2022-12-31 00:00:00 | 2 | 0.6082192 |

## Getting to the Right Answer

In real data, we don't know what the true rate is. We have to estimate it from the information we are given. In insurance experience studies, I have seen the ratio of deaths to exposures used for the mortality rate. Doing this is almost harmless for very rare death rates (think younger ages). I have also seen the more strict ratio of deaths to the quantity of exposures plus half the deaths. This is arguably more correct, and the analyst needs to decide for themselves what they need to do.

Unfortunately, the ratio of deaths to exposures describes (cumulative) hazard and not a probability. For very small death rates, they are close. For larger death rates, they are rather different. Moveover, the hazard rate can validly be greater than 1. The formula connecting cumulative hazard $\mu$ and mean mortality rate $q$ is

$$q = 1 - e^{-\mu}$$

Let's see what happens when mortality rates are quite high. Note that this issue affects lapse studies as well, where high rates are much more common.

### Overall

```
census %>%
  inner_join(y=exposures,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date == exp_period_end,
                            1,
                            0),
         ExpectedDeathsUsingHazard=-exposure*log(1-annual_rate),
         ExpectedDeathsUsingPaddedExposure=(exposure+Death_Count/2)*annual_rate) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Hazard=Death_Count/sum(exposure),
            RateFromHazard=1-exp(-Hazard),
            RateFromPadding=Death_Count/(Exposure+Death_Count/2),
            A_E_Hazard=Death_Count/sum(ExpectedDeathsUsingHazard),
            A_E_Padded=Death_Count/sum(ExpectedDeathsUsingPaddedExposure),
            S_E=1/sqrt(sum(Death_Count))
  ) %>%
  flextable() %>%
  colformat_num(
    j=c("Death_Count","Exposure")
  ) %>%
  colformat_double(
    j="Hazard",
    digits=3
  ) %>%
  set_formatter(
    RateFromHazard = function(x) scales::percent(x,accuracy=.01),
    RateFromPadding = function(x) scales::percent(x,accuracy=.01),
    A_E_Hazard  = function(x) scales::percent(x,accuracy=.01),
    A_E_Padded = function(x) scales::percent(x,accuracy=.01),
    S_E = function(x) scales::percent(x,accuracy=.01)
  ) %>%
  set_header_labels(
    Death_Count="Death Count",
    RateFromHazard="Mortality Rate From Hazard",
    RateFromPadding="Mortality Rate From Padding",
    A_E_Hazard= "A/E - Hazard Based",
    A_E_Padded = "A/E - Padding Based",
    S_E = "Standard Error"
  ) %>%
  autofit() %>%
```

```
  fit_to_width(
    max_width = 6
  )
```

| Death Count | Exposure | Hazard | Mortality Rate From Hazard | Mortality Rate From Padding | A/E - Hazard Based | A/E - Padding Based | Standard Error |
|---|---|---|---|---|---|---|---|
| 65,308 | 94,062.68 | 0.694 | 50.06% | 51.54% | 100.17% | 103.08% | 0.39% |

Using exact exposure, the rate from the hazard is almost exactly what we would expect to see. Padding the deaths is apparently inadequate here. The hazard-based estimate is well within 1 standard error, while the padded calculation is well beyond it. This would be a margin for protection coverage and an underestimate for annuities.

## By Duration

By policy duration and calendar year, the hazards-based estimate remains solid, while the estimate based on padding continues to be materially inaccurate.

```
census %>%
  inner_join(y=exposures,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date == exp_period_end,
                            1,
                            0),
         ExpectedDeathsUsingHazard=-exposure*log(1-annual_rate),
         ExpectedDeathsUsingPaddedExposure=(exposure+Death_Count/2)*annual_rate) %>%
  group_by(pol_duration) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Hazard=Death_Count/sum(exposure),
            RateFromHazard=1-exp(-Hazard),
            RateFromPadding=Death_Count/(Exposure+Death_Count/2),
            A_E_Hazard=Death_Count/sum(ExpectedDeathsUsingHazard),
            A_E_Padded=Death_Count/sum(ExpectedDeathsUsingPaddedExposure),
            S_E=1/sqrt(sum(Death_Count))
  ) %>%
  flextable() %>%
  colformat_num(
    j=c("Death_Count","Exposure")
  ) %>%
  colformat_double(
```

```
    j="Hazard",
    digits=3
  ) %>%
  set_formatter(
    RateFromHazard = function(x) scales::percent(x,accuracy=.01),
    RateFromPadding = function(x) scales::percent(x,accuracy=.01),
    A_E_Hazard  = function(x) scales::percent(x,accuracy=.01),
    A_E_Padded = function(x) scales::percent(x,accuracy=.01),
    S_E = function(x) scales::percent(x,accuracy=.01)
  ) %>%
  set_header_labels(
    pol_duration="Policy Duration",
    Death_Count="Death Count",
    RateFromHazard="Mortality Rate From Hazard",
    RateFromPadding="Mortality Rate From Padding",
    A_E_Hazard= "A/E - Hazard Based",
    A_E_Padded = "A/E - Padding Based",
    S_E = "Standard Error"
  ) %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| Policy Duration | Death Count | Exposure | Hazard | Mortality Rate From Hazard | Mortality Rate From Padding | A/E - Hazard Based | A/E - Padding Based | Standard Error |
|---|---|---|---|---|---|---|---|---|
| 1 | 44,926 | 64,555.808 | 0.696 | 50.14% | 51.63% | 100.40% | 103.26% | 0.47% |
| 2 | 16,720 | 24,254.126 | 0.689 | 49.81% | 51.27% | 99.45% | 102.53% | 0.77% |
| 3 | 3,662 | 5,252.742 | 0.697 | 50.20% | 51.70% | 100.58% | 103.39% | 1.65% |

## By Calendar Year

```
census %>%
  inner_join(y=exposures,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date == exp_period_end,
                            1,
                            0),
         ExpectedDeathsUsingHazard=-exposure*log(1-annual_rate),
         ExpectedDeathsUsingPaddedExposure=(exposure+Death_Count/2)*annual_rate) %>%
```

```
  group_by(Year=year(exp_period_end)) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Hazard=Death_Count/sum(exposure),
            RateFromHazard=1-exp(-Hazard),
            RateFromPadding=Death_Count/(Exposure+Death_Count/2),
            A_E_Hazard=Death_Count/sum(ExpectedDeathsUsingHazard),
            A_E_Padded=Death_Count/sum(ExpectedDeathsUsingPaddedExposure),
            S_E=1/sqrt(sum(Death_Count))
  ) %>%
  flextable() %>%
  colformat_num(
    j=c("Death_Count","Exposure")
  ) %>%
  colformat_double(
    j="Hazard",
    digits=3
  ) %>%
  set_formatter(
    RateFromHazard = function(x) scales::percent(x,accuracy=.01),
    RateFromPadding = function(x) scales::percent(x,accuracy=.01),
    A_E_Hazard  = function(x) scales::percent(x,accuracy=.01),
    A_E_Padded = function(x) scales::percent(x,accuracy=.01),
    S_E = function(x) scales::percent(x,accuracy=.01)
  ) %>%
  set_header_labels(
    Year="Calendar Year",
    Death_Count="Death Count",
    RateFromHazard="Mortality Rate From Hazard",
    RateFromPadding="Mortality Rate From Padding",
    A_E_Hazard= "A/E - Hazard Based",
    A_E_Padded = "A/E - Padding Based",
    S_E = "Standard Error"
  ) %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| Calendar Year | Death Count | Exposure | Hazard | Mortality Rate From Hazard | Mortality Rate From Padding | A/E - Hazard Based | A/E - Padding Based | Standard Error |
|---|---|---|---|---|---|---|---|---|
| 2,021 | 26,024 | 37,503.81 | 0.694 | 50.04% | 51.52% | 100.11% | 103.03% | 0.62% |
| 2,022 | 28,565 | 41,121.89 | 0.695 | 50.07% | 51.56% | 100.22% | 103.11% | 0.59% |
| 2,023 | 10,719 | 15,436.98 | 0.694 | 50.06% | 51.54% | 100.18% | 103.08% | 0.97% |

## Padding the Underlying Exposures

Some study designs pad the underlying exposures directly. That is, the exposure record itself is extended to an entire policy duration, regardless of when in the policy duration the death occurred.

To check this situation, we will

- change the death date to coincide with the day before the policy anniversary
- expand the exposures
- restore the death date to the exact death date
- trim incomplete records

```
# Create the census of policies, including simulated dates
copy(census) %>%
  mutate(  Death_Date=Issue_Date %m+% days( ceiling(events/365)*365 - 1 )) ->
  census_padded_deaths

# Set the Termination Date, which is the earlier of the Death Date and
# Lapse Date, unless the dates are after the study.
census_padded_deaths %>%
  mutate(Term_Date = as_datetime(
    ifelse(Death_Date <= Lapse_Date & Death_Date <= study_end,
           Death_Date,
           NA))) %>%
  mutate(Term_Date = as_datetime(
    ifelse(Lapse_Date <= Death_Date & Lapse_Date <= study_end,
           Lapse_Date,
           Term_Date))) ->
  census_padded_deaths


cacheFile <- file.path(
  cacheRoot,
  "measuring-risk-exposures-padded-deaths.parquet"
```

```
)

if(bUseCache &
   !bInvalidateCache &
   file.exists(
     cacheFile
   )
) {
  exposures_padded_deaths <- arrow::read_parquet(
    cacheFile
  )
} else {
  census_padded_deaths %>%
    filter(Issue_Date <= study_end) %>%
    select(PolID,Issue_Date,Term_Date,Prem_Mode_Months) %>%
    expand_exposures(
      .exp_period_start = study_start,
      .exp_period_end = study_end,
      .cal_yr_breaks = cal_yr_breaks,
      .pol_period_granularity = pol_period_granularity,
      .cal_period_granularity = cal_period_granularity,
      .issue_date = Issue_Date,
      .term_date = Term_Date,
      .ID=PolID,
      .prem_mode_months = Prem_Mode_Months,
      .exact_terminations = FALSE
    ) ->
    exposures_padded_deaths

  census_padded_deaths %>%
    filter(is.na(Term_Date)) %>%
    inner_join(
      y=exposures_padded_deaths
    ) %>%
    group_by(PolID,
             pol_duration) %>%
    filter(sum(exposure)==1) %>%
    ungroup() %>%
    data.table() %>%
    union_all(
      y=census_padded_deaths %>%
        inner_join(
```

```
          y=exposures_padded_deaths
        ) %>%
        filter(!is.na(Term_Date))
    ) %>%
    select(
      -Issue_Date,
      -Prem_Mode_Months,
      -Death_Date,
      -Lapse_Date,
      -Term_Date
    ) ->
    exposures_padded_deaths

  if(bUseCache) {
    arrow::write_parquet(
      x=exposures_padded_deaths,
      sink=cacheFile
    )
  }
}

# Reset the death date

cacheFile <- file.path(
       cacheRoot,
       "measuring-risk-census-padded-deaths.parquet"
     )

if(bUseCache &
   !bInvalidateCache &
   file.exists(
     cacheFile
   )) {
  census_padded_deaths <- arrow::read_parquet(
    cacheFile
  )
} else {
  # Create the census of policies, including simulated dates
  census_padded_deaths %>%
    mutate(  Death_Date=Issue_Date %m+% days(events),
             Lapse_Date=Issue_Date %m+% months(events2)) ->
    census_padded_deaths
```

```
  # Set the Termination Date, which is the earlier of the Death Date and
  # Lapse Date, unless the dates are after the study.
  census_padded_deaths %>%
    mutate(Term_Date = as_datetime(
      ifelse(Death_Date <= Lapse_Date & Death_Date <= study_end,
             Death_Date,
             NA))) %>%
    mutate(Term_Date = as_datetime(
      ifelse(Lapse_Date <= Death_Date & Lapse_Date <= study_end,
             Lapse_Date,
             Term_Date))) ->
    census_padded_deaths

  if(bUseCache)
    arrow::write_parquet(
      x=census_padded_deaths,
      sink=cacheFile
    )
}
```

We see immediately that there are issues with this approach. The bias remains, with rates and A/E ratios well outside the envelope of uncertainty. We also see some unpleasant issues. By calendar year, there appears to be too little exposure in the first year, and too much in later years. Padding out the exposure records has distorted the estimates by pushing "dangling" exposures into future years. Things look better by policy duration, but the final duration has the same issue as calendar year, with inadequate exposures beyond the current calendar year being excluded.

```
census_padded_deaths %>%
  inner_join(y=exposures_padded_deaths,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date %between% list(
    exp_period_start,
    exp_period_end),
                            1,
                            0),
         ExpectedDeathsUsingExposure=(exposure)*annual_rate) %>%
  #group_by(Year=year(exp_period_end)) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Rate=Death_Count/(Exposure),
            A_E=Death_Count/sum(ExpectedDeathsUsingExposure),
```

```
            S_E=1/sqrt(sum(Death_Count))
  ) %>%
  flextable() %>%
  colformat_num(
    j=c("Death_Count","Exposure")
  ) %>%
  set_formatter(
    Rate= function(x) scales::percent(x,accuracy=.01),
    A_E  = function(x) scales::percent(x,accuracy=.01),
    S_E = function(x) scales::percent(x,accuracy=.01)
  ) %>%
  set_header_labels(
    #Year="Calendar Year",
    Death_Count="Death Count",
    Rate="Mortality Rate",
    A_E= "A/E",
    S_E = "Standard Error"
  ) %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| Death Count | Exposure | Mortality Rate | A/E | Standard Error |
|---|---|---|---|---|
| 65,223 | 123,605.2 | 52.77% | 105.53% | 0.39% |

```
census_padded_deaths %>%
  inner_join(y=exposures_padded_deaths,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date %between% list(
    exp_period_start,
    exp_period_end),
                            1,
                            0),
         ExpectedDeathsUsingExposure=(exposure)*annual_rate) %>%
  group_by(Year=year(exp_period_end)) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Rate=Death_Count/(Exposure),
            A_E=Death_Count/sum(ExpectedDeathsUsingExposure),
            S_E=1/sqrt(sum(Death_Count))
```

```
  ) %>%
  flextable() %>%
  colformat_num(
    j=c("Death_Count","Exposure")
  ) %>%
  set_formatter(
    Rate= function(x) scales::percent(x,accuracy=.01),
    A_E  = function(x) scales::percent(x,accuracy=.01),
    S_E = function(x) scales::percent(x,accuracy=.01)
  ) %>%
  set_header_labels(
    Year="Calendar Year",
    Death_Count="Death Count",
    Rate="Mortality Rate",
    A_E= "A/E",
    S_E = "Standard Error"
  ) %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| Calendar Year | Death Count | Exposure | Mortality Rate | A/E | Standard Error |
|---|---|---|---|---|---|
| 2,021 | 26,024 | 46,081.53 | 56.47% | 112.95% | 0.62% |
| 2,022 | 28,508 | 59,187.11 | 48.17% | 96.33% | 0.59% |
| 2,023 | 10,691 | 18,336.51 | 58.30% | 116.61% | 0.97% |

```
census_padded_deaths %>%
  inner_join(y=exposures_padded_deaths,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date %between% list(
    exp_period_start,
    exp_period_end),
                            1,
                            0),
         ExpectedDeathsUsingExposure=(exposure)*annual_rate) %>%
  group_by(pol_duration) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Rate=Death_Count/(Exposure),
```

```
            A_E=Death_Count/sum(ExpectedDeathsUsingExposure),
            S_E=1/sqrt(sum(Death_Count))
) %>%
flextable() %>%
colformat_num(
  j=c("Death_Count","Exposure")
) %>%
set_formatter(
  Rate= function(x) scales::percent(x,accuracy=.01),
  A_E  = function(x) scales::percent(x,accuracy=.01),
  S_E = function(x) scales::percent(x,accuracy=.01)
) %>%
set_header_labels(
  pol_duration="Policy Duration",
  Death_Count="Death Count",
  Rate="Mortality Rate",
  A_E= "A/E",
  S_E = "Standard Error"
) %>%
autofit() %>%
fit_to_width(
  max_width = 6
)
```

| Policy Duration | Death Count | Exposure | Mortality Rate | A/E | Standard Error |
|---|---|---|---|---|---|
| 1 | 44,869 | 87,968.307 | 51.01% | 102.01% | 0.47% |
| 2 | 16,692 | 32,946.460 | 50.66% | 101.33% | 0.77% |
| 3 | 3,662 | 2,690.386 | 136.11% | 272.23% | 1.65% |

## Pitfalls for the Naive

The simple situation above suggests that exact exposures are the right approach. I have seen experience studies implemented this way. However, care must be taken when comparing against a baseline mortality and how ratios are treated

A common approach includes setting expected claims equal to exposure times the mortality rate and to use death claims divided by exposure for the raw mortality rate. For small mortality rates, this is fine. However, for larger mortality rates, this will lead to the wrong answer.

### Illustrating the Problem

```
census %>%
  inner_join(y=exposures,by=join_by(PolID)) %>%
  mutate(Death_Count=ifelse(Death_Date == exp_period_end,
                            1,
                            0),
         ExpectedDeathsUsingHazard=-exposure*log(1-0.4),
         ExpectedDeathsExposure=(exposure)*0.4) %>%
  summarize(Death_Count=sum(Death_Count),
            Exposure=sum(exposure),
            Naive_Death_Rate=Death_Count/Exposure,
            True_Death_Rate=1-exp(-Death_Count/Exposure),
            A_E_Naive=Death_Count/sum(ExpectedDeathsExposure),
            A_E_Hazard=sum(Death_Count)/sum(ExpectedDeathsUsingHazard),
            A_E_RateScale=True_Death_Rate/(1-exp(-sum(ExpectedDeathsUsingHazard)/Exposure)),
            S_E=1/sqrt(sum(Death_Count))
  ) %>%
  flextable() %>%
  colformat_num(
    j=c("Death_Count","Exposure")
  ) %>%
  colformat_double(
    j="Naive_Death_Rate",
    digits=3
  ) %>%
  set_formatter(
    Naive_Death_Rate = function(x) scales::percent(x,accuracy=.01),
    A_E_Naive = function(x) scales::percent(x,accuracy=.01),
    A_E_Hazard= function(x) scales::percent(x,accuracy=.01),
    A_E_RateScale= function(x) scales::percent(x,accuracy=.01),
    S_E = function(x) scales::percent(x,accuracy=.01)
  ) %>%
  set_header_labels(
    Death_Count="Death Count",
    Naive_Death_Rate="Naive Death Rate",
    True_Death_Rate="True Death Rate",
    A_E_Naive = "A/E - Naive Expected",
    A_E_Hazard = "A/E - Hazards",
    A_E_RateScale="A/E - q scale",
    S_E = "Standard Error"
  ) %>%
```

```
autofit() %>%
fit_to_width(
  max_width = 6
)
```

| Death Count | Exposure | Naive Death Rate | True Death Rate | A/E - Naive Expected | A/E - Hazards | A/E - q scale | Standard Error |
|---|---|---|---|---|---|---|---|
| 65,308 | 94,062.68 | 69.43% | 0.5005776 | 173.58% | 135.92% | 125.14% | 0.39% |

If you were expecting 0.4 and did not account for the fact that you were using a hazards ratio to approximate a mortality rate, you would be left quite concerned. You may not be aware that there are methodological issues in your study. Moreover, the downstream impact would be obvious in that your claims assumption would not backcast historical patterns accurately.

The reason for this is that in the world of exact exposure, you either need to transform the hazards ratio to a mortality rate and compare it to the expected claims or accept that you should model hazards directly.

When modeling the hazards, the ratio between the actual hazard and baseline hazard will translate into an exponent on the survivor probability as follows:

$$q_{modeled} = 1 - (1 - q_{baseline})^r$$

If you expand the right hand side, you see the following:

$$

$$
\begin{aligned}
1 - (1 - q_{baseline})^r &= 1 - \sum_{k=0}^{\infty} \binom{r}{k} (-q_{baseline})^k \\
&= 1 - \left[1 - rq_{baseline} + \frac{r(r-1)}{2!} q_{baseline}^2 - \frac{r(r-1)(r-2)}{3!} q_{baseline}^3 + \cdots\right] \\
&= rq_{baseline} - \frac{r(r-1)}{2!} q_{baseline}^2 + \frac{r(r-1)(r-2)}{3!} q_{baseline}^3 + \cdots
\end{aligned}
$$

$$

The $rq_{baseline}$ term should be familiar, since that is all you need for small rates. But for large r and q, the higher terms become significant. In our example where $r$ is 135.92% and the baseline $q$ is 0.4, we need four terms of the expansion to get to three decimal places (0.50074).

On the other hand, that can be a lot of hassle. Transforming both the actual hazards and baseline hazards to the probability scale yields the more familiar actual-to-expected ratio of 125.14%, which will recover the true mortality rate when applied to 0.4.

### When Should You Care

I have stated without evidence that for small values of $r$ and $q_{baseline}$, there is no issue using the product $rq_{baseline}$.

The binomial expansion above has error dominated by the second term. I have provided a table which computes the ratio of the second term to the true mortality rate under this model.

```
expand_grid(
  q_baseline=c(0.5,0.4,0.25,0.1,.05,0.01,0.001,0.0001),
  r=seq(.4,2,length.out=9)
) %>%
  mutate(
    approx_error=scales::percent(1-r*q_baseline/(1-(1-q_baseline)^r),
                                 accuracy=.1)
  ) %>%
  pivot_wider(names_from=q_baseline,
              values_from = approx_error) %>%
  flextable() %>%
  set_formatter(
    r=function(x) scales::percent(x,accuracy=1)
  ) %>%
  align(
    align="right",
    part="all"
  ) %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| r | 0.5 | 0.4 | 0.25 | 0.1 | 0.05 | 0.01 | 0.001 | 1e-04 |
|---|---|---|---|---|---|---|---|---|
| 40% | 17.4% | 13.4% | 8.0% | 3.1% | 1.5% | 0.3% | 0.0% | 0.0% |
| 60% | 11.8% | 9.1% | 5.4% | 2.1% | 1.0% | 0.2% | 0.0% | 0.0% |
| 80% | 6.0% | 4.6% | 2.7% | 1.0% | 0.5% | 0.1% | 0.0% | 0.0% |
| 100% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% | 0.0% |
| 120% | -6.2% | -4.7% | -2.8% | -1.0% | -0.5% | -0.1% | 0.0% | 0.0% |
| 140% | -12.7% | -9.6% | -5.6% | -2.1% | -1.0% | -0.2% | 0.0% | 0.0% |
| 160% | -19.4% | -14.6% | -8.4% | -3.1% | -1.5% | -0.3% | 0.0% | 0.0% |

| r | 0.5 | 0.4 | 0.25 | 0.1 | 0.05 | 0.01 | 0.001 | 1e-04 |
|---|---|---|---|---|---|---|---|---|
| 180% | -26.3% | -19.7% | -11.3% | -4.2% | -2.0% | -0.4% | 0.0% | 0.0% |
| 200% | -33.3% | -25.0% | -14.3% | -5.3% | -2.6% | -0.5% | -0.1% | 0.0% |

Whether these potential errors are important depends on the application. It appears that baseline rates at or above 1-5% are where problems may be found.

- Mortality: In the 2015 VBT ultimate table for male non-smokers (ANB), the mortality rate of 1% corresponds approximately to age 69.
- DI Morbidity: Incidence and termination rates can readily fall into the range of 1-10%, based on a quick inspection of the 2013 IDI tables from the Society of Actuaries.
- LTC Morbidity: Initial terminations excluding deaths appear to lie in the 1-5% range, while incidence rates are above 1% for policy years 10+ (and potentially elsewhere depending on subset). This is based on a high level review of the 2000-2016 LTC aggregate files from the Society of Actuaries.
- Lapses: Lapses, especially initial lapses, can be quite high. Depending on target market, they can be in the 20-45% range.

## Basic Predictive Models

When we introduce predictive models into the mix, getting this right matters. Traditionally, survival modelings have relied on Cox proportional-hazards models due to their flexibility. They are semi-parametric, combining a parametric model overlaid on a non-parametric baseline hazards. However, Cox models demand that the proportional hazards assumption be satisfied, which is typically not true in life insurance mortality and many other life and health applications. Therefore, equivalent GLMs are used in place of Cox models to handle the complexity of life insurance mortality.

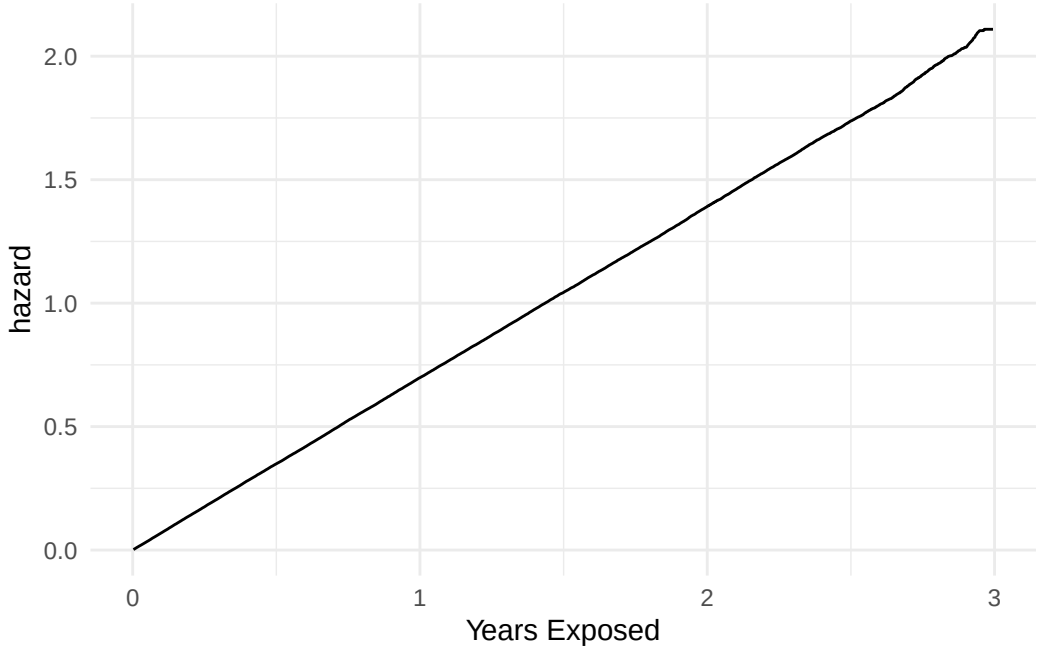
In the first case where mortality was 0.5, we can recover the true rate by either using Cox models or GLMs.

In the plot below, we see the cumulative hazards from a Cox model fit to the first example. It is a straight line, and there are ways to obtain the hazard and hence the mortality rate

```
library(survival)

census %>%
  mutate(
    Days_Exposed=interval(Issue_Date,
                          ifelse(is.na(Term_Date),study_end,Term_Date)
```

```
                          ) / years(1),
    Status=ifelse(Term_Date == Death_Date & !is.na(Term_Date),1,0)
  ) %>%
  coxph(
    formula=Surv(Days_Exposed,Status) ~1,
    data=.
  ) %>%
  basehaz() %>%
  data.table() %>%
  ggplot(aes(x=time,y=hazard)) +
  geom_line() +
  scale_x_continuous(name="Years Exposed") +
  theme_minimal()
```




To do so, we can just take differences in the cumulative hazard to get the within period cumulative hazard, and then apply the transform to get a mortality rate. The hazard curve suffers somewhat from noise at the end of the study.

```
census %>%
  mutate(
    Days_Exposed=interval(Issue_Date,
                          ifelse(is.na(Term_Date),study_end,Term_Date)
    ) / years(1),
```

```
    Status=ifelse(Term_Date == Death_Date & !is.na(Term_Date),1,0)
  ) %>%
  coxph(
    formula=Surv(Days_Exposed,Status) ~1,
    data=.
  ) %>%
  basehaz() %>%
  data.table() ->
  cph.basehaz

union(
  x=cph.basehaz %>% filter(time %in% c(1,2)),
  y=cph.basehaz %>% slice_tail(n=1)
) %>%
  mutate(period_cml_hazard=hazard-lag(hazard,default=0),
         time_diff=time-lag(time,default=0)) %>%
  mutate(qx=1-exp(-period_cml_hazard/time_diff)) %>%
  select(-hazard,-time_diff) %>%
  flextable() %>%
  colformat_double(
    j=c("period_cml_hazard","qx","time"),
    digits=3
  ) %>%
  set_header_labels(
    time="Ending Policy Duration",
    period_cml_hazard="Period Total Hazard"
  ) %>%
  autofit() %>%
  fit_to_width(
    max_width = 6
  )
```

| Ending Policy Duration | Period Total Hazard | qx |
|---:|---:|---:|
| 1.000 | 0.698 | 0.502 |
| 2.000 | 0.693 | 0.500 |
| 2.995 | 0.718 | 0.514 |

GLMs can also recover the true value. Unlike a Cox model, exposure is required along with understanding what is being modeled.

If the response variable is death count, and the offset is log of exposure, then the GLM is modeling the hazard rate. If the family is Poisson with log link, one gets the original rate as the coefficient in this case.

```
census %>%
  inner_join(
    y=exposures
  ) %>%
  mutate(
    Death_Count=ifelse(Death_Date %between%
                         list(
                           exp_period_start,
                           exp_period_end
                         ),1,0)
  ) %>%
  filter(exposure > 0) %>%
  glm(
    formula=Death_Count ~ 1,
    data=.,
    family=quasipoisson(link="log"),
    offset=log(exposure)
  ) %>%
  flextable::as_flextable()
```

```
Joining with `by = join_by(PolID)`
```

| | Estimate | Standard Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -0.365 | 0.008 | -44.669 | 0.0000 *** |

Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05

(Dispersion parameter for quasipoisson family taken to be 4.357048)

Null deviance: 2.609e+05 on 241026 degrees of freedom

Residual deviance: 2.609e+05 on 241026 degrees of freedom

The coefficient here is the log of the hazard.

If on the other hand I want to model the mortality rate directly, then the offset must include the some adjustment for censoring.

```
census %>%
  inner_join(
    y=exposures
  ) %>%
  mutate(
    Death_Count=ifelse(Death_Date %between%
                         list(
                           exp_period_start,
                           exp_period_end
                         ),1,0)
  ) %>%
  filter(exposure > 0) %>%
  glm(
    formula=Death_Count ~ 1,
    data=.,
    family=quasipoisson(link="log"),
    offset=log(exposure + Death_Count/2)
  ) %>%
  flextable::as_flextable()
```

```
Joining with `by = join_by(PolID)`
```

| | Estimate | Standard Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | -0.663 | 0.003 | -255.518 | 0.0000 *** |

Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05

(Dispersion parameter for quasipoisson family taken to be 0.4394842)

Null deviance: 1.214e+05 on 241026 degrees of freedom

Residual deviance: 1.214e+05 on 241026 degrees of freedom

The resulting estimate is the log of the mortality rate we saw before in the padded case, which can be readily recalled as 0.515.

I can easily put my expected hazard as an offset.

```r
census %>%
  inner_join(
    y=exposures
  ) %>%
  mutate(
    Death_Count=ifelse(Death_Date %between%
                         list(
                           exp_period_start,
                           exp_period_end
                         ),1,0),
    ExpectedClaimHazard = -exposure*log(1-0.4)
  ) %>%
  filter(exposure > 0) %>%
  glm(
    formula=Death_Count ~ 1,
    data=.,
    family=quasipoisson(link="log"),
    offset=log(ExpectedClaimHazard)
  ) %>%
  flextable::as_flextable()
```

```
Joining with `by = join_by(PolID)`
```

| | Estimate | Standard Error | z value | Pr(>\|z\|) |
|---|---|---|---|---|
| (Intercept) | 0.307 | 0.008 | 37.572 | 0.0000 *** |

Signif. codes: 0 <= '***' < 0.001 < '**' < 0.01 < '*' < 0.05

(Dispersion parameter for quasipoisson family taken to be 4.357048)

Null deviance: 2.609e+05 on 241026 degrees of freedom

Residual deviance: 2.609e+05 on 241026 degrees of freedom

This is the log of the ratio of actual hazard to expected hazard.