Analysis of Count & Recurrent Events
Week 11

PH 700A, Spring 2025

Rick Calvo

Table of contents

1 Week 11 Overview

1.1 Welcome Back!

Spring break was last week. :(

Today's content will be more for your reference vs. immediate application.

Primary focus will be on *Poisson Regression*.

Remaining Schedule

Week	Topic
11	Analysis of "Count" Data
12	Data Reduction
13	Dimensionality Reduction
14	Predictive Modeling
15	Visualization w/ ggplot2
16	Finals Week (No Meeting)

1.2 Session Outline

- Packages
- Methods Overview
- Andersen-Gill

- Poisson Regression
- Generalized Estimating Equations & Mixed Models
- Example Analysis

1.3 Packages

1.3.1 Andersen-Gill Survival Analysis

```
library(survival)
```

1.3.2 Poisson Regression

```
library(stats)
library(multcomp)
library(broom)
```

1.3.3 Generalized Estimating Equations

```
library(gee)
```

1.3.4 Mixed-effects Modeling

```
library(lme4)
```

1.4 A Note on Diagnostics

Usage of advanced methods requires advance validation of prerequisites and assumptions. There are no *post hoc* diagnostics.

Decision to use an advanced analysis should be based on *bivariate analyses*, *linear*, *logistic*, and standard *proportional hazards* model results.

Packages will seldom contain functions to assess variable characteristics such as outliers, normality, etc…

Utility of higher-level analytical techniques will hinge on *your satisfaction* of the assumptions prior to analysis.

2 Analysis of Non-Singular Outcomes

2.1 Background

- Health outcomes do not necessarily have to be finite with time
- Not all health outcomes are singular events
- Events/outcomes of *the same type* can be analyzed as:
  - Count-based
  - Recurrent
  - Repeated
- This is different from *cumulative* outcomes, which would typically be analyzed as a linear or ordinal process
- *Logistic* and *Cox* regression can usually only address the *first* occurrence of an event

2.2 And...?

2.3 Importance

- Prior events can affect risk for subsequent events
  - Violates the *assumption of independence*
- Additional events may imply a different risk level vs. singular events
  - Risk factor associations for the first event may differ with subsequent events
- The first event and subsequent events are not necessarily identical
  - Prior events may be catalysts for future events
- Intervals of time between events may vary
  - Temporal framework may "reset" at each experience of an event
  - Risk factors may vary with time (i.e. smoker at baseline may have quit some time later)

2.4 Analytic Options

1. Andersen-Gill Proportional Hazards Model
   - Recurrent binary events over time
2. Poisson Regression
   - Counts of events within a time interval
3. Generalized Estimating Equations
   - Longitudinal analysis with periodic measurement
   - Assumes *low heterogenity* of sample
4. Mixed-effects Model
   - Longitudinal analysis with periodic measurement
   - Addresses *high heterogeneity* of sample

3 Andersen-Gill Proportional Hazards Model

3.1 Purpose

- An extension of the Cox Proportional Hazards Model
- For the analysis of recurrent events among participants of a study
- Participants may have several *start* and *stop* times denoting separate "windows" of observation
- Independent variables can change values over time and are called *time-varying covariates*

3.2 Assumptions

- *Some amount* of correlation between prior and subsequent events exists
- The amount of time *between events* is not correlated; It is only that the events are correlated
- Changes to *risk factor* status may *vary with time* (a *time-varying covariate*) and may explain the correlation between events
- The *proportional hazards assumption* still applies
- The `coxzph()` function can still be used to evaluate the assumption
- In the presence of *time-varying covariates*, the proportional hazards assumption will appear violated

3.3 Schema

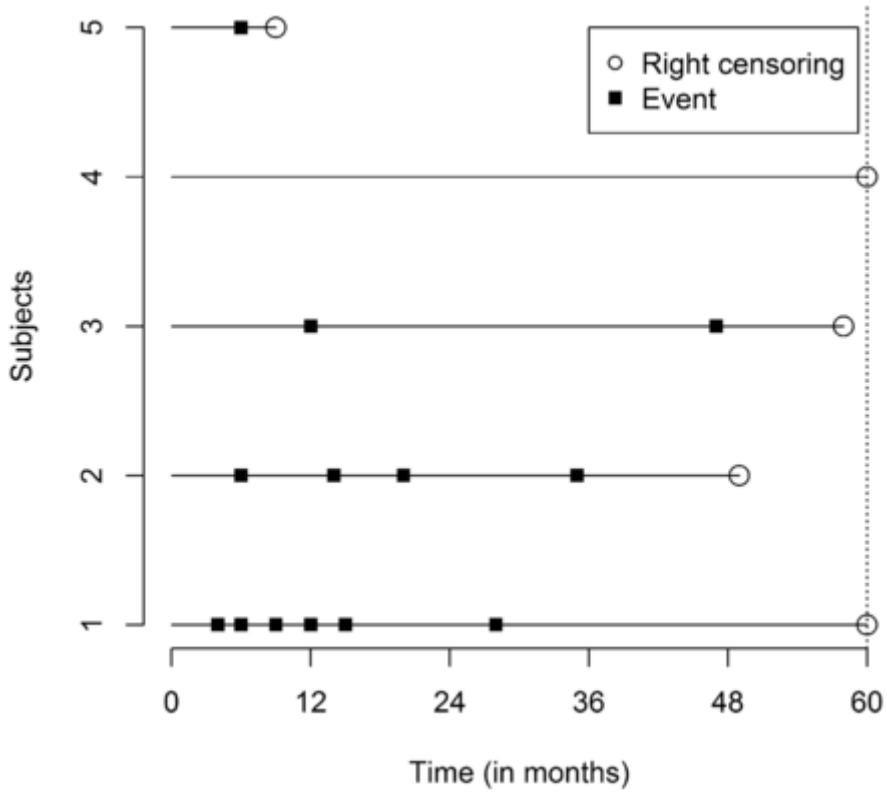


Figure 1: Timeline of five hypothetical patients. From Amorim, 2015.

Employing `start` and `stop` times for each observation allows us to treat time in a stratified manner.

If `stop` times are not available, time will have to be manually stratified. (`library(survSplit)`; See *Zhang, 2018* for an example.)

3.4 Data Requirements

Table 2: Records for 5 hypothetical patients. Adapted from *Amorim, 2015*.

idvar	start	stop	event	count	trt	var1	...
A	0	1	1	1	0	1	...
A	3	7	1	2	0	4	...
B	0	4	0	0	0	2	...
B	5	7	0	0	1	1	...
C	0	10	1	1	1	5	...
C	14	16	0	1	1	4	...
D	0	8	0	0	1	1	...
E	0	18	0	0	0	1	...
E	22	25	1	1	0	1	...
E	26	31	1	2	1	2	...
E	32	36	0	2	1	2	...

3.5 Commands

```
library(survival)

model1 <- coxph(Surv(start, stop, event) ~ trt + ... + varn + cluster(idvar),
                method = "efron",
                robust = TRUE,
                data = df)

cox.zph(model1)

summary(model1)
```

- `cluster(idvar)` specifies the unique identifier variable
- `method =` tells R how to handle ties. Options include: `breslow`, `efron`, `exact`. If there are no ties, results will be the same
  - **breslow** is the simplest; the default for other programs
  - **efron** has higher accuracy; the default for R
  - **exact** is the most accurate but computationally intensive
- `robust =`, when equal to **TRUE**, calculates robust variance estimates (via "jackknife" resampling)
  - Required for long data; automatically **TRUE** if the formula contains `cluster()`

4 Poisson Regression

4.1 Purpose

To evaluate relationships between *risk factors* and *event incidence rates*.

- Note that we are not modeling the occurrence of the event itself

Permits for the adjustment of confounders/interactions that may affect incidence rate changes

If the event of interest is rare, identifying factors related to the *population incidence rate* is more reasonable than factors for the event.

In essence:

- There will be some amount of incidence that "just happens" with time
- We evaluate if a risk factor is accountable for some of the incidence vs. how much "just happens"

4.2 Comparison to Other Methods

Linear, logistic, and Poisson are all within the family of **Generalized Linear Models**.

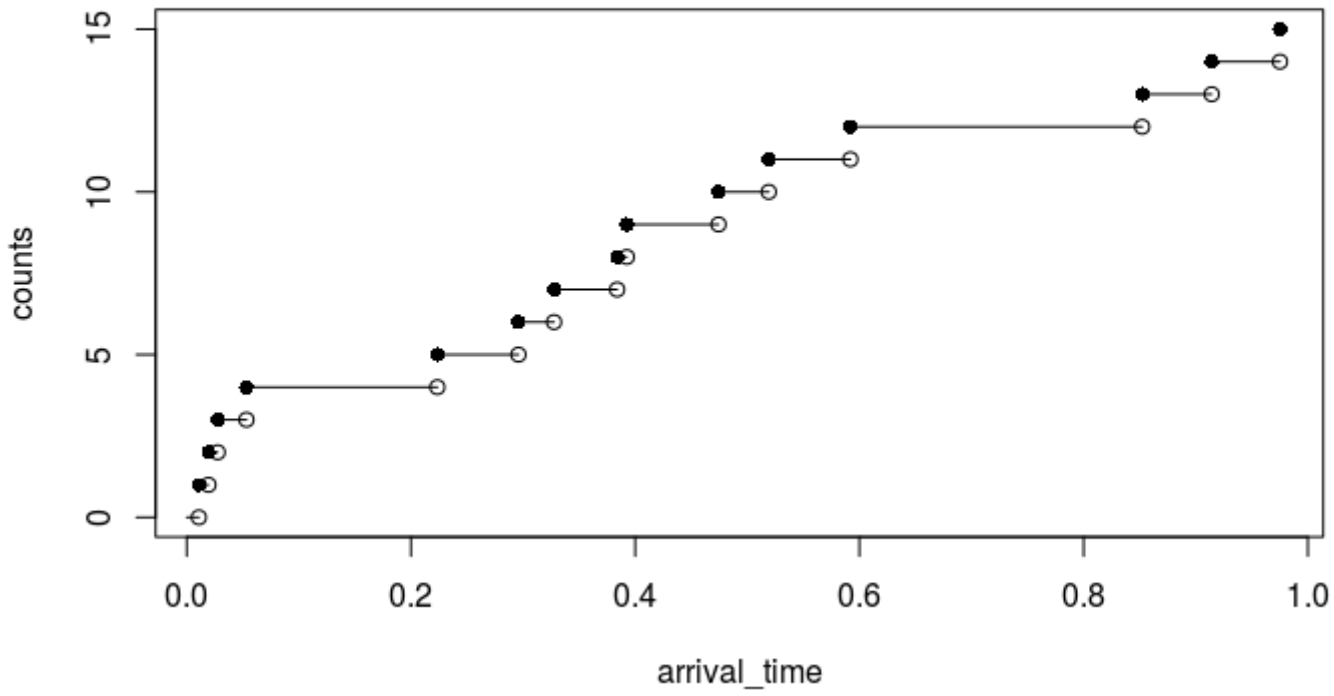
Linear regression is not appropriate for *counts* because:

- The minimum value you can have for the outcome is 0 and the maximum is INF
- Transformation of the variable would result in an unrealistic interpretation
- Equality of variances assumption is impossible to achieve
- Linear regression results in a linear function; Poisson regression results in a log function

Ordinal logistic regression is not appropriate for *counts* because:

- Categorization of the count outcome results in a loss of information
- Categorization can be biased and/or violate the proportional odds assumption
- The interval between each count value is static whereas ordered categories do not a preexisting interval of difference
- The error distribution for logistic regression is binomial rather than a Poisson distribution

4.3 Schema



The image demonstrates events (solid circle) occurring over time (arrival time). The line shows the duration of time elapsed until another event is experienced.

There is no discernible pattern in the way events occur.

4.4 Assumptions

- There is **no correlation** between events or the timing of the event, and events occur at a standard rate over time
- There is context to the event occurrence – either by *time*, *space*, or both
- The mean of the event counts must also be equal to the variance
- Measurement of the event follows a Bernoulli distribution – either the event happens or it does not
- Only one event can occur at any given time
- The probability of the event occurring is proportional to the amount of time spent observing it

4.5 Data Requirements

Poisson regression does not *directly* account for follow-up time because individual risk is not the focus.

idvar	start	stop	event	count	trt	var1	...
A	0	1	1	1	0	1	...
A	3	7	1	2	0	4	...
B	0	4	0	0	0	2	...
B	5	7	0	0	1	1	...
C	0	10	1	1	1	5	...
C	14	16	0	1	1	4	...
D	0	8	0	0	1	1	...
E	0	18	0	0	0	1	...
E	22	25	1	1	0	1	...
E	26	31	1	2	1	2	...
E	32	36	0	2	1	2	...

Long data needs to be converted to wide data.

Relevant factors need to be "collapsed" during conversion to wide.

4.6 Collapsing Data

The process of converting individual-level data to aggregates by *some grouping* method.

`library(tidyverse)`

`summarise()`

Individual-level Data

idvar	start	stop	event	count	doses	sbp	...
A	0	1	1	1	3	100	...
A	3	7	1	2	7	110	...
B	0	4	0	0	0	95	...
B	5	7	0	0	6	105	...

Collapsed data

idvar	time	events	tot_doses	avg_sbp	...
A	7	2	10	105	...
B	7	0	6	100	...

Collapsing will always result in *some loss* of information.

4.7 Poisson Commands

The standard `glm()` function can be modified to perform a Poisson regression.

```
model1 <- glm(eventcount ~ var1,
              family = poisson,
              data = df)

summary(model1)
  exp(coef(model1))
  exp(confint(model1))

model2 <- glm(eventcount ~ var1 + var2,
              family = poisson,
              data = df)

summary(model2)
  exp(coef(model2))
  exp(confint(model2))

devianceTest <- anova(model1, model2, test = "Chisq")

summary(devianceTest)
```

By default, models will only contain raw estimates.

Therefore, must manually exponentiate the coefficients and confidence intervals to get an interpretable *Incidence Rate Ratio*.

The *Incidence Rate Ratio* is just a ratio of incidence rates between two groups.

4.8 Model Evaluation and Development

Model development should be iterative and thoughtful.

4.8.1 Internal Validity

- The *Wald test* statistic (z^2) can be used to test if the variables in the model are contributing to explanation of the event
- Effective "explanation" of the event incidence by model covariates will result in a $p < 0.050$

4.8.2 Variable Selection

- Changes in *deviance* between models can be used to guide variable selection
- A *reduction of deviance* implies better fit, termed *drop-in-deviance*
- A standard `anova` is used to evaluate a statistically significant *drop-in-deviance*
- If $p < 0.050$, then the *modified model* has a statistically significant improvement in fitness and should be retained

4.9 Offset Option

Although time is a collapsed variable, it is still important.

- The ability to observe an event is dependent on how long we watch
- Must account for differences in observation time to get a sense of how "bad" the incidence actually is

```
model <- glm(counts ~ var1 + ... + varn,
             family = poisson,
             offset = log(time),
             data = df)`
```

4.10 Multiple Comparisons Testing

Two packages `multcomp` and `broom` help facilitate multiple comparisons tests for Poisson regression.

Multiple comparisons tests of the IRRs are done with the *Tukey* method.

```
library(multcomp)
library(broom)

model.mc1 <- summary.glht(model, mcp(var1="Tukey"))
```

- `model` is the object containing the `glm()` results of the Poisson regression
- `summary.glht()` is a function that stands for "Generalized Linear Hypothesis Testing"
  - Uses the results from `glm()` and creates a summary tibble
- `mcp()` specifies a multiple comparisons analysis using the "Tukey" method for the stated variable
  - `var1` is a *factor* variable specified in the model

5 Analysis of Repeated Measures Data (Panel Data)

5.1 Purpose

Data are longitudinally ascertained at intervals over a study period where the outcome and exposure are each measured.

Outcomes and exposures can vary over time and we are interested in identifying associations between them.

Two main methods:

- *Generalized Estimating Equations* (marginal model)
- *Mixed-effects Regression* (conditional model)

Correlation between measurements must be accounted for; Results from one time can influence those at a subsequent time.

5.2 Expanding Your Toolkit

GEE and *mixed models* are not specifically for the analysis of repeated measures data.

Generally speaking, these models can account for situations where the *assumption of independence* cannot be confidently stated.

Application to repeated measures data addresses *loss of independence* due to correlation of multiple measurements of the same individual over time

However, *GEE* and *mixed models* can be used for *any scenario* with internal correlation by specifying the organizational *data structure*.

5.3 Schema - Clustering of Participants

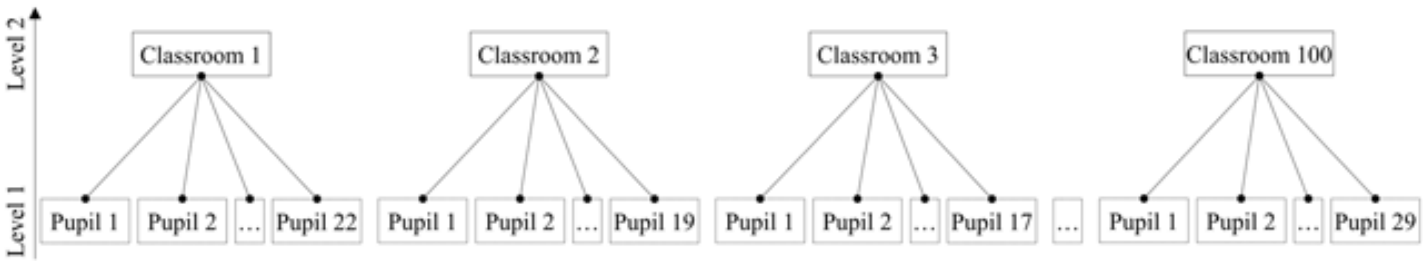


Figure 2: Clustering Example; From Sommet, 2017

5.4 Schema - Repeated Measures Across Participants

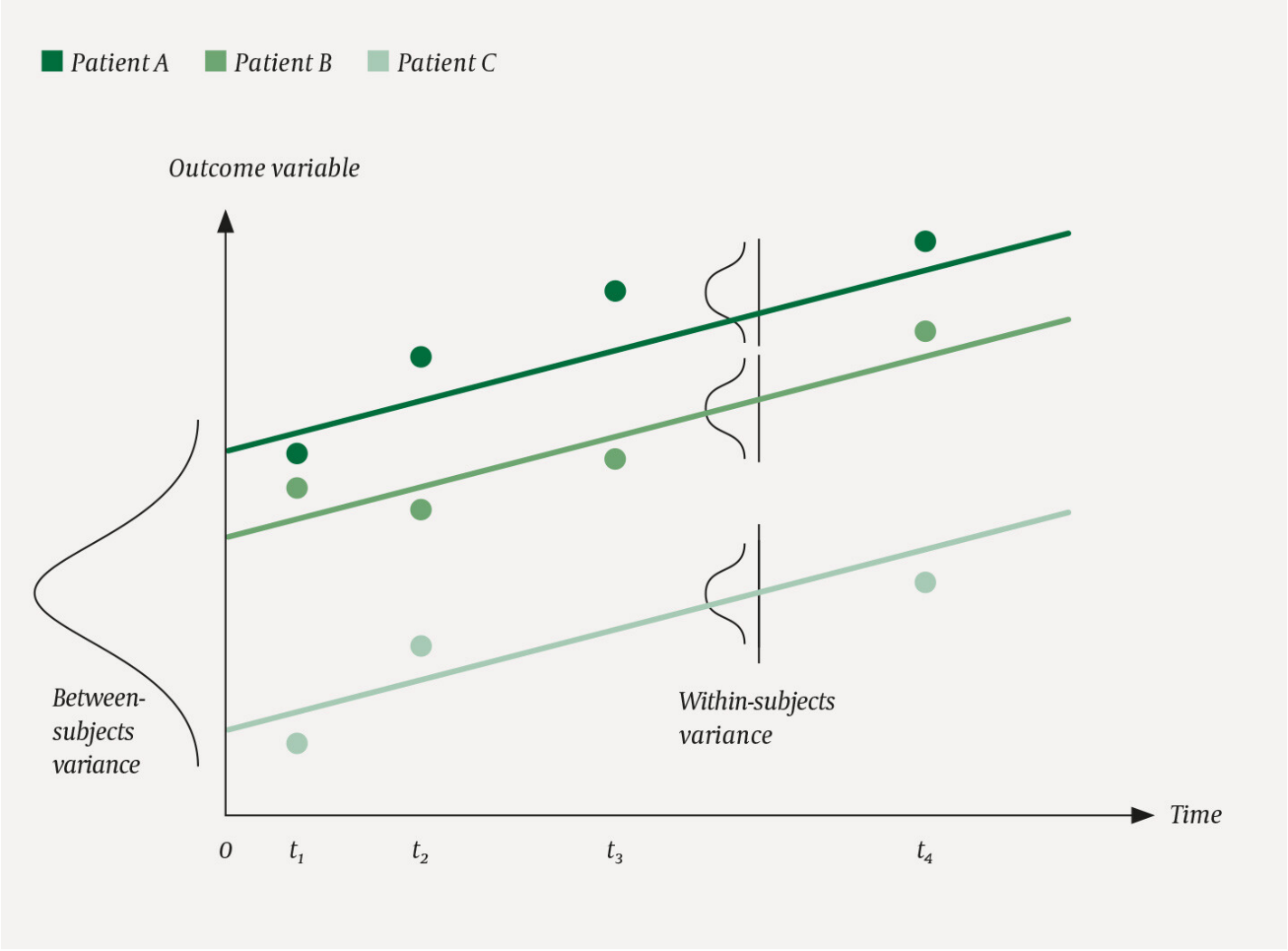


Figure 3: Events over time by individual; From Lyderson 2022

5.5 Primary Assumptions

- Data are missing completely at random
- Variances and means are not equal to each other
- There is a structure to the covariance that can explain the correlation across measurements
- The correlation can be "captured" by an organizing variable in the data frame

5.6 Data Requirements

pt	time	statu	measu	var1	var2	var3	...
1	0	0	50	1	0	1	...
1	4	1	90	2	0	4	...
1	10	1	95	1	0	2	...
2	0	0	70	1	1	1	...
2	3	0	79	1	1	5	...
3	0	1	101	2	1	4	...
3	4	0	85	1	1	1	...
4	0	1	110	1	0	1	...
5	0	1	95	1	0	1	...
5	6	1	80	2	0	2	...
5	11	0	90	3	0	2	...

Note that `time` is a single variable to denote *when* the measurements occurred. This is not an interval.

5.7 Comparison of Model Types

Both model types address issues with *internal correlation* via repeated measurement.

Generalized Estimating Equations are **marginal models**

- Results pertain to a *population average*
- No likelihood function; estimation only
- One set of variables where:
  - *time* is evaluated as an interaction
  - *participant id* is specified "outside" the formula
- No correlation due to *structural factors*

Mixed effects models are **conditional models**

- Results are applied to *event risk/probability for individuals*
- Models are fit using "Restricted" MLE
- Contains a list of *fixed* and *random* effects that are estimated differently
  - *fixed* effects are specific to the patient/unit of observation
  - *random* effects are grouping variables that may be common among individuals or measurements
- Correlation due to internal and external factors are *controlled for* via random effects

5.8 Commands (GEE)

Package `gee` contains the commands for *Generalized Estimating Equations* and can be used for continuous and categorical outcomes.

```
library(gee)

# Categorical Outcome
catgee <- gee(statu ~ var1*time + var2 + var3*time, data = df,
              id = pt,
              family = binomial,
              corstr = "independence")

catgee$working.correlation

# Continuous Outcome
contgee <- gee(measu ~ var1*time + var2 + var3*time, data = df,
              id = pt,
              family = gaussian,
              corstr = "independence")

contgee$working.correlation
```

- `var1*time` is a `var1` * `time` interaction denoting that `var1` is a time-varying covariate
  - `var3*time` is also evaluated as a time-varying covariate
- `id = pt` tells R how the data organized, where `pt` is the patient identification number in long-form data
- `family = binomial` states that a binomial distribution will be used for modeling (i.e. a dichotomous outcome).
  - `family = gaussian` could be used for continuous outcomes
- `corstr = "independence"` identifies the covariance structure (the structure of correlation between measurements). Options are:
  - `"independence"` states that there is actually no correlation between measurements
  - `"exchangeable"` states that within-subject correlation between each pair of measurements is equal
  - `"AR-M", Mv = 1` identifies a first-order autoregressive correlation, that measurements become more correlated over time
  - `"unstructured"` makes no assumption on the structure and will compute every correlation. This is computationally intensive

5.9 Commands (Mixed Models)

Package `lme4` contains the commands for *Mixed-Effects Models* and can be used for continuous and categorical outcomes.

```
library(lme4)

# Logistic Mixed Model
logmix <- glmer(statu ~ var1 + var2 + var3*time + (1 | pt), data = df, family = binomial, nAGQ = 1,

summary(logmix, corr = FALSE)

# Linear Mixed Model
linmix <- glmer(measu ~ var1 + var2 + var3*time + (1 | pt), data = df, family = gaussian, nAGQ = 1,

summary(linmix, corr = FALSE)
```

- `glmer()` is the function to generate a mixed-effects model
- `var1 + var2 + var3*time` are fixed effects. `var3*time` evaluates `var3` as a time-varying covariate
- `(1 | pt)` specifies `pt` as a random effect
  - Since `pt` is a random effect, every person in the study gets their own intercept (i.e. a unique starting point)
- `nAGQ = 1`: The *Adaptive Gauss-Hermite Quadrature* is a modified likelihood used to fit the model. The default number of points is 1
  - **Quadrature points**: Origins in calculus; Can specify # of dimensions for calculating approximations for a solution. Higher is more accurate but computationally intense
- `control = glmerControl(optimizer = "bobyqa")`: Applies an optimization algorithm to improve calculations
  - `bobyqa`: *Bound Optimization By Quadratic Approximation*

6 Poisson Analysis Example

6.1 Data Preparation

A patient-level analysis of risk factors for incidence of tachycardic events.

Tachycardia is defined as a heart rate exceeding 100 beats per minute.

A *tachycardic event* is defined as a sustained heart rate exceeding 100 bpm for over 30 seconds.

```
df.tach %>%
  select(stay_id, charttime, temperature, heartrate, resprate, o2sat, sbp, dbp, tachycardia) %>%
  head()
```

```
# A tibble: 6 x 9
   stay_id charttime           temperature heartrate resprate o2sat   sbp   dbp
     <int> <dttm>                    <dbl>     <int>    <int> <int> <int> <int>
1 30094124 2147-11-14 18:23:00        97.3        80       18    99   138    78
2 30094124 2147-11-14 22:08:00        98.5        82       18   100   142    78
3 30115213 2147-12-30 08:40:00        NA          83       16   100   132    70
4 30115213 2147-12-30 08:48:00        97.2        89       17   100   118    64
5 30193781 2198-04-17 11:42:00        98.2        88       16   100   150    98
6 30193781 2198-04-17 15:09:00        97.8       105       18    97   173    97
# i 1 more variable: tachycardia <dbl>
```

```
head(df.tach.sum)
```

```
# A tibble: 6 x 9
   stay_id t0                  t1                  avg_temp avg_rr avg_o2sat
     <int> <dttm>              <dttm>                 <dbl>  <dbl>     <dbl>
1 30094124 2147-11-14 18:23:00 2147-11-14 22:08:00     97.9   18        99.5
2 30115213 2147-12-30 08:40:00 2147-12-30 08:48:00     97.2   16.5     100
3 30193781 2198-04-17 11:42:00 2198-04-17 19:48:00     99.3   17.6      97.6
4 30225689 2149-09-20 05:50:00 2149-09-20 15:54:00     97.8   16.4      99.2
5 30272878 2131-05-22 20:34:00 2131-05-22 21:13:00     97.4   19.8      96.5
6 30279522 2149-09-17 09:10:00 2149-09-17 23:00:00     98.0   17.7      98.4
# i 3 more variables: avg_sbp <dbl>, avg_dbp <dbl>, tachycardia <dbl>
```

```
#rm(df.tach.sum, df.tach)
```

The `df2` data frame contains only the patients with at least one record in `df.vitalsign` and was created from an `inner_join` with `df`.

A variable, `elapsed`, is the difference between `t1` and `t0` from the vital signs measurements. This is the total interval of time (in seconds) that vital signs data were collected.

Values of 0 in `elapsed` occur if only one measurement was collected. This was converted to 1 second to resolve issues with logarithmic transformation.

`tachycardia` is a count of total tachycardic events identified from `df.vitalsign`.

Other vital signs measurements (`temperature`, `resprate`, `o2sat`, `sbp`, `dbp`) were averaged over the vital signs time interval.

6.2 Model Construction

```
library(gtsummary)

tach.m1 <- glm(tachycardia ~ racewhite, family = poisson, offset = log(elapsed), data = df2)
summary(tach.m1)
```

Characteristic	IRR	95% CI	p-value
racewhite			
0	—	—	
1	1.70	1.07, 2.77	0.028

Abbreviations: CI = Confidence Interval, IRR = Incidence Rate Ratio

```
Call:
glm(formula = tachycardia ~ racewhite, family = poisson, data = df2,
    offset = log(elapsed))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.4150     0.2000  -57.08   <2e-16 ***
racewhite1    0.5292     0.2405    2.20   0.0278 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 270.52  on 205  degrees of freedom
Residual deviance: 265.40  on 204  degrees of freedom
AIC: 413.14

Number of Fisher Scoring iterations: 6
```

```
tach.m1 %>% tbl_regression(exponentiate = TRUE)
```

```
tach.m2 <- glm(tachycardia ~ racewhite + ambulance, family = poisson, offset = log(elapsed), data =

summary(tach.m2)
```

```
Call:
glm(formula = tachycardia ~ racewhite + ambulance, family = poisson,
    data = df2, offset = log(elapsed))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -11.3231     0.2385 -47.467   <2e-16 ***
racewhite1    0.5442     0.2416   2.253   0.0243 *
ambulance1   -0.1579     0.2311  -0.683   0.4944
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Characteristic	IRR	95% CI	p-value
racewhite			
0	—	—	
1	1.72	1.09, 2.81	0.024
ambulance			
0	—	—	
1	0.85	0.55, 1.36	0.5

Abbreviations: CI = Confidence Interval, IRR = Incidence Rate Ratio

```
(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 270.52  on 205  degrees of freedom
Residual deviance: 264.94  on 203  degrees of freedom
AIC: 414.68

Number of Fisher Scoring iterations: 7
```

```
tach.m2 %>% tbl_regression(exponentiate = TRUE)
```

```
tach.1v2 <- anova(tach.m1, tach.m2, test = "Chisq")

tach.1v2
```

```
Analysis of Deviance Table

Model 1: tachycardia ~ racewhite
Model 2: tachycardia ~ racewhite + ambulance
  Resid. Df Resid. Dev Df Deviance Pr(>Chi)
1       204     265.40
2       203     264.94  1   0.4604   0.4974
```

Drop-in-deviance is only 0.4604 and this was not stat sig w/ p = 0.4974.

Elect not to retain `ambulance`.