



# Introduction to Statistics in R

Workshop 2 – June 24, 2025





## Introduction to Statistics in R: Part 2

A two-part course on using R for statistical analysis for SickKids researchers and trainees.

Topics for Part 2 include:

- Generalized linear models
- Multi-level modeling
- Data visualization

Prerequisites: familiarity with RStudio and linear regressions

To register please RSVP at:  
<https://ccm20250624.eventbrite.com>

Organized by the Centre for Computational Medicine ([ccm.sickkids.ca](http://ccm.sickkids.ca)) at the SickKids Research Institute with support from other groups:



[www.sickkids.ca/research](http://www.sickkids.ca/research)



Digital Research  
Alliance of Canada

[alliancecan.ca](http://alliancecan.ca)



Compute  
Ontario

[computeontario.ca](http://computeontario.ca)

## Hands-on Bioinformatics Tutorials for Biologists

Tuesday June 24, 2025  
9 AM – 12 PM

In person in Multimedia Room,  
PGCRL 3<sup>rd</sup> floor, or on Zoom



# CCM Overview

Note that prior knowledge of R is recommended to get the most out of this workshop series.

Workshop 1	Data exploration and introductory statistics
Workshop 2	Generalized and hierarchical regressions

Check out other workshop series hosted by CCM at  
<https://ccm.sickkids.ca/bioinformatics-training/>



# Table of contents

**01**

## **Recap of linear regression**

Overview of contrasts

**02**

## **Generalized linear models**

Logistic, Poisson, and Zero-Inflated Poisson regression

**03**

## **Multi-level models**





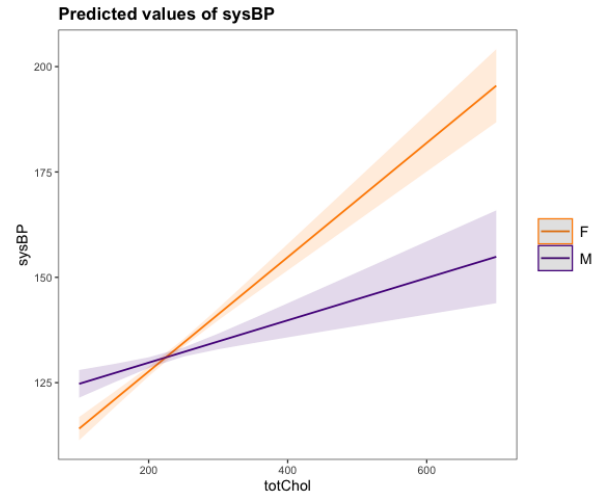
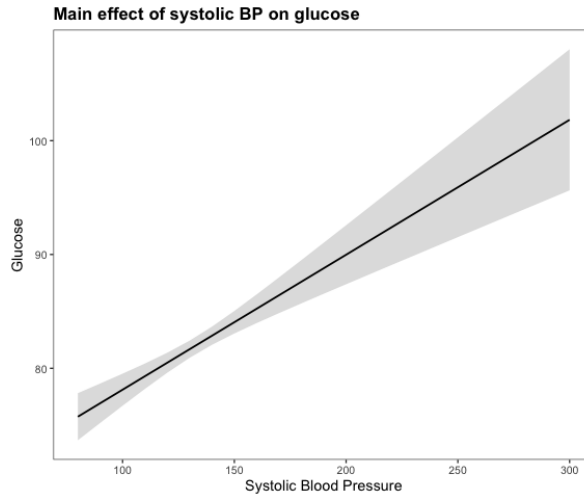
01

# Recap of linear regression and introduction to contrasts

# Linear Regression

Model relationships between predictors and continuous outcome variables.

- Main effects models reveal the linear relationship between each predictor and the outcome.
- Interaction effects reveal how the relationship between a predictor and an outcome is *moderated* by another variable.



# Linear Regression

Example of a main effects model:

```
lm(cholesterol_levels ~ sex + BP_meds, data = dataframe)
```

Example of an interaction effect model:

```
lm(outcome ~ predictor1 * predictor2, data = df)
```

Syntax	Description
<code>outcome ~ predictor1 + predictor2 + ...</code>	Main effects of predictor 1, predictor 2, etc.
<code>outcome ~ predictor1 + predictor2 + predictor1:predictor2</code>	Main effects of predictor 1 and predictor 2, and the interaction between predictor 1 and 2
<code>outcome ~ predictor1 * predictor2</code>	Main effects of predictor 1 and predictor 2, and the interaction between predictor 1 and predictor 2
<code>outcome ~ predictor1 + predictor1:predictor2</code>	Main effect of predictor 1 only, and the interaction between predictor 1 and predictor 2
<code>outcome ~ predictor1:predictor2</code>	The interaction between predictor 1 and predictor 2 only



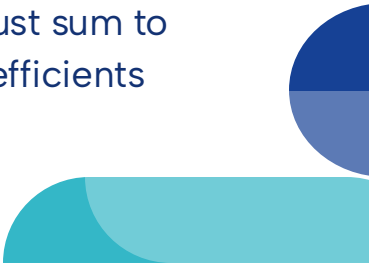
# Contrasts

Contrasts are linear combinations used to represent categorical predictors in a regression model. For a categorical predictor with  $n$  levels, you need  $n - 1$  contrasts to encode the variable.

There are several types of contrasts:

1. Dummy coding
2. Simple effect coding
3. Deviation coding
4. Difference coding
5. Helmert coding

For all contrast types except for dummy coding, the contrast coefficients must sum to 0. This centers the predictor around the mean, making the intercept and coefficients easier to interpret.



# Contrasts

By default, R uses **dummy coding** (aka treatment contrasts) where the reference group is the *first level* of the factor (this can be changed) and represented with 0.

The intercept is the mean of the reference group. The resulting coefficients represent the difference between the level and the reference level.

`contr.treatment(n)`

vegetables	Contrast 1	Contrast 2
Never	0	0
Sometimes	1	0
Always	0	1

Gender	Contrast 1
Female	0
Male	1



# Contrasts

With **simple effect coding**, you can compare each level to a manually-defined reference level (coded as -1). The intercept represents the grand mean of the outcome variable across all levels, while the coefficients represent the difference between a given level and the reference. Must be specified manually in R!

```
contrasts(df$var) <- cbind(contr1 = c(-1, 1, 0),  
                           contr2 = c(-1, 0, 1))  
contrasts(df$var) <- c(-1, 1)
```

vegetables	Contrast 1	Contrast 2
Never	-1	-1
Sometimes	1	0
Always	0	1
Sum	0	0

Gender	Contrast 1
Female	-1
Male	1
Sum	0

# Contrasts

**Deviation coding** compares each level to the *grand mean* instead of a reference group. The intercept represents the grand mean while the coefficients represent the difference between the level and the grand mean. There's no reference group.

`contr.sum(n)`

vegetables	Contrast 1	Contrast 2
Never	-0.25	-0.25
Sometimes	0.50	-0.25
Always	-0.25	0.50
Sum	0	0

# Contrasts

**Difference coding** is useful for comparing the mean of the current level to the mean of the previous levels. This is useful when you have ordinal variables that are ordered in a meaningful way (e.g., low, medium, high).

**Helmert coding** is useful for comparing the mean of the current level to the mean of the *subsequent* levels. Essentially, the opposite of difference coding.

vegetables	Contrast 1	Contrast 2
Never	-1	-0.5
Sometimes	1	-0.5
Always	0	1
Sum	0	0

Difference Coding

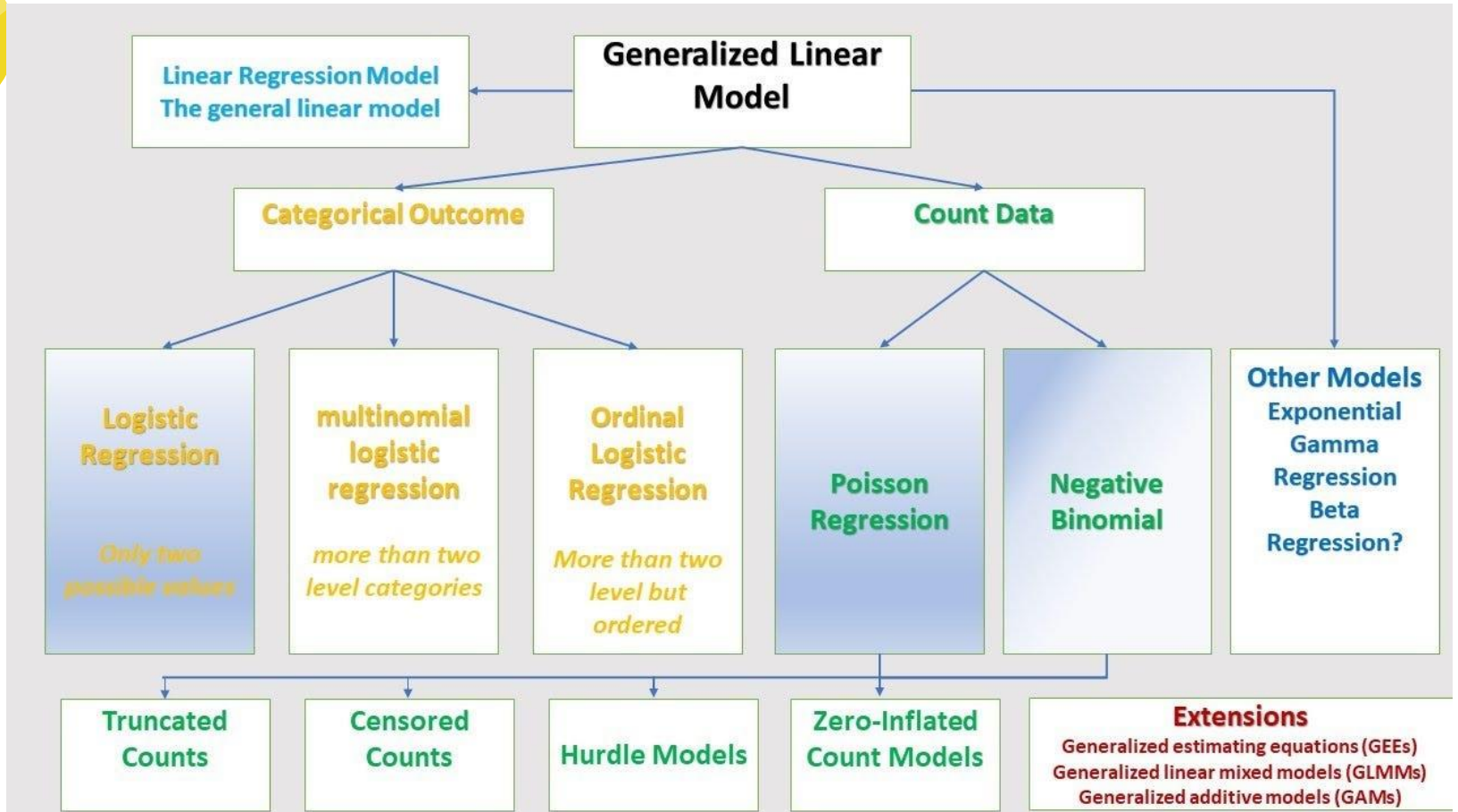
vegetables	Contrast 1	Contrast 2
Never	-1	-1
Sometimes	1	-1
Always	0	2
Sum	0	0

Helmert Coding



02

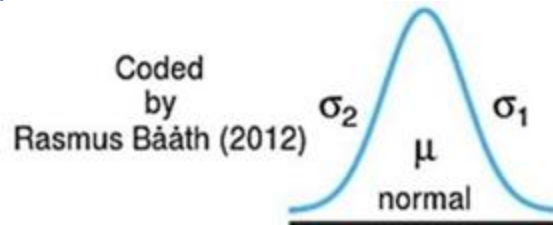
# Generalized linear models



# Generalized Linear Models (GLMs)

Linear regressions assume that the residuals of the outcome variable are normally distributed i.e., they follow the Gaussian distribution.

This assumption generally holds well for most continuous variables.



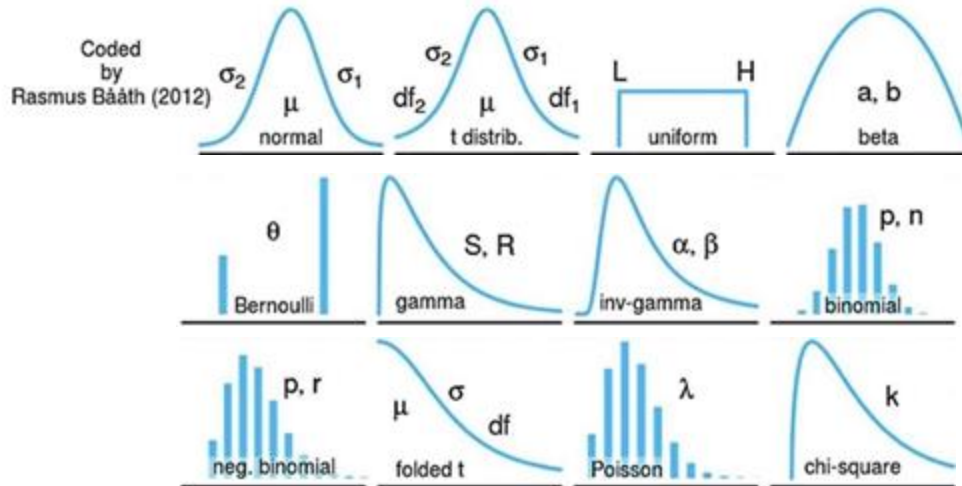
In practice, many forms of data do not meet this assumption.

- In some cases, you can transform the variable (e.g., log transformation to reduce skewness) to make the data more normally distributed (e.g., income).
- In other cases, you will need to use a model that assumes that your outcome comes from a non-Gaussian distribution family.

# Generalized Linear Models (GLMs)

GLMs are used with data types like binary variables, counts, and proportions.

The outcome variable typically comes from a non-normal/Gaussian distribution, like the binomial (e.g., Bernoulli for binary variables), Poisson (e.g., for counts), beta (e.g., for proportions) and so on.



# Generalized Linear Models (GLMs)

Link functions connect (or “link”) the linear predictors to the expected value of the outcome variable’s distribution, often by transforming the outcome to a scale where linear modeling is appropriate.

Distribution	Common Link Function	Purpose/Usage	Formula
Normal	Identity	Directly relates the linear predictor to the response variable. Used for continuous data.	$\eta = \mu$
Binomial	Logit	Transforms the probability of success to an unbounded scale. Used for binary outcomes (e.g., success/failure).	$\eta = \log\left(\frac{\mu}{1-\mu}\right)$
Poisson	Log	Relates the log of the mean count to the linear predictors. Used for count data.	$\eta = \log(\mu)$



# Framingham Heart Study

Examinations included:

- Detailed medical history
- Physical exams
- Lab tests
- Lifestyle and habits
- Noninvasive imaging

Our dataset: subset of the FHS from Kaggle

- 4000+ records and 16 variables.

<https://www.kaggle.com/datasets/captainozlem/framingham-chd-preprocessed-data>

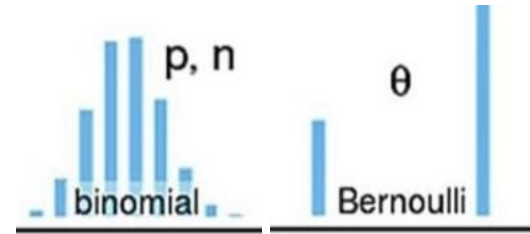
Variable	Description	Class/type
male	Sex (male or female)	Binary
age	Age	Continuous
education	Education (0-11 years, HS or GED, some uni, uni grad+)	Ordinal/categorical
currentSmoker	Whether the patient is currently a smoker	Binary
cigsPerDay	Number of cigarettes smoked per day, on average	Continuous
BPMeds	Whether the patient is on blood pressure medication	Binary
prevalentStroke	Whether the patient previously had a stroke	Binary
prevalentHyp	Whether the patient is hypertensive	Binary
diabetes	Whether the patient has diabetes	Binary
totChol	Total cholesterol level	Continuous
sysBP	Systolic blood pressure	Continuous
diaBP	Diastolic blood pressure	Continuous
BMI	Body mass index	Continuous
heartRate	Heart rate	Continuous
glucose	Glucose level	Continuous
TenYearCHD	10-year risk of coronary heart disease	Binary

# Logistic Regression

**When to use?** When your outcome variable is binary (e.g., yes/no, true/false, case/control, diagnosis A vs diagnosis B).

**Distribution:** binomial distribution.

**Link function:** logit (log-odds).



The coefficients/estimates from the model output are logits and are not easily interpretable, but can be converted to odds ratios:  $e^{logit}$ .

- Odds ratios indicate the odds of your outcome variable occurring as you increase your predictor by 1 unit (for continuous variables) or go from one group to the next (for categorical variables).



# Logistic Regression

Let's look at the relationship between smoking status, sex, and heart disease.

```
log_model <- glm(TenYearCHD ~ sex * currentSmoker, data = df, family =  
  binomial(link="logit"))  
  
exp(coef(log_model))
```

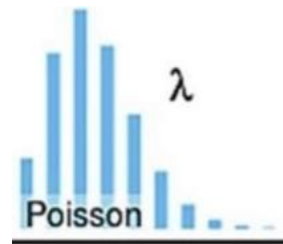


# Poisson Regression

**When to use?** When your outcome variable is a count and the mean is approximately equal to the variance. This approach is especially suitable for rare events (e.g., fewer than 10 counts).

**Distribution:** Poisson.

**Link function:** natural log ( $\ln$ ).



The coefficients/estimates from the model output are on the log scale and represent the log of the expected count. They can be exponentiated to obtain rate ratios:  $e^{\beta}$ .



# Poisson Regression

Let's look at the relationship between lifestyle factors and physiological factors.

```
poisson_model <- glm(biometric_flags ~ age * sex + education +  
cigsPerDay + diabetes, data = df, family = poisson(link="log"))
```

```
exp(coef(poisson_model))
```



# Zero-Inflated Poisson Regression

**When to use?** When your outcome variable is a count, but there are more 0's in your data than expected.

**Distribution:** Poisson, Bernoulli.

**Link function:** log, logit.



The zero-inflated Poisson (ZIP) model assumes that zero counts can arise from:

1. A Bernoulli process where a datapoint can belong to the always-zero group or not.
2. A Poisson process where a datapoint can take on a value of zero or a count.



# Zero-Inflated Poisson Regression

Let's look at the relationship between smoking status, sex, and heart disease.

```
zip_model <- zeroinfl(cigsPerDay ~ age * sex | age * sex,  
                      data = df,  
                      dist = "poisson")
```

The first `age * sex` term models the Poisson process (counts) and the second `age * sex` term models the Bernoulli process (zero-inflation).





# Other GLMs

## Negative binomial models

- Generalization of Poisson regression, when the **count data are over-dispersed** i.e., their variance is greater than the mean. Uses the log link function and coefficients represent the log of expected counts. Coefficients can be exponentiated to get rate ratios.

## Multinomial regression

- Generalization of logistic regression, when there are **>2 unordered categories** in the outcome variable. Uses the generalized logit link function and coefficients represent the change in log-odds of being in one category compared to the reference category. As with logistic regression, coefficients can be exponentiated to get odds ratios.

## Ordinal regression

- Generalization of logistic regression, when there are **>2 ordered categories** in the outcome variable. Uses the cumulative logit link function and coefficients represent the change in log-odds of being in a higher vs lower category. Coefficients can be exponentiated to get cumulative odds ratios.

## Beta regression

- Used when the outcome variable is a **proportion between 0 and 1**, exclusive. Uses a logit or log link function and coefficients represent the change in the mean proportion.
- 





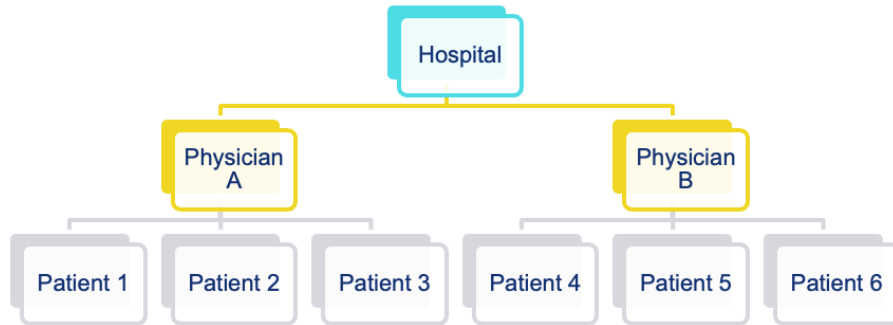
03

# Multi-level models

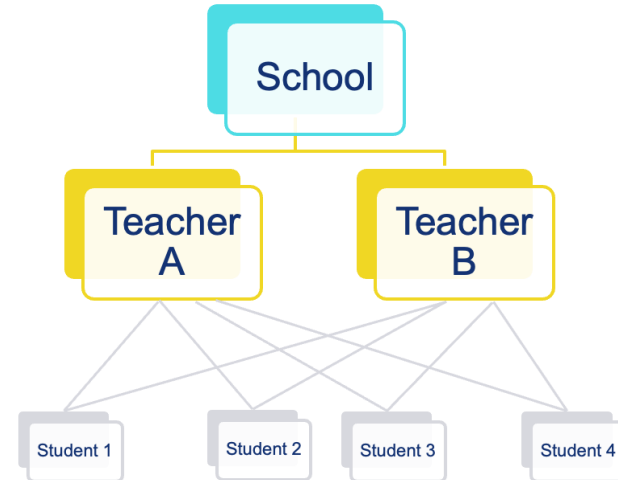
# Multi-Level Models

Multi-level models can be useful when:

- Standard linear model assumptions are violated.
- Data are unbalanced.
- Observations are systematically grouped or nested within a hierarchy.



Nested Data



Crossed Data



# Fixed and Random Effects

Multi-level models allow you to model fixed and random effects, which is why they are also referred to as mixed-effects models.


**Fixed effects:** model the population-level average effect.

- Same predictors you would include in your linear or generalized regression models.
- Assumed to be constant across individuals.
- Interpretation is the effect of predictor X on your outcome Y, averaged across the whole sample.

**Random effects:** model the variability in data not captured by the fixed effects.

- How individual units deviate from the overall average.
- Common levels:
  - Level 1 = smallest unit of observation (e.g., student, trial, patient).
  - Level 2 = grouping level that stays constant across level 1 (e.g., teacher, participant, doctor).

Random effects allow you to generalize beyond the specific (level 2) groups in your sample, thereby improving inference and accounting for dependency in the data.





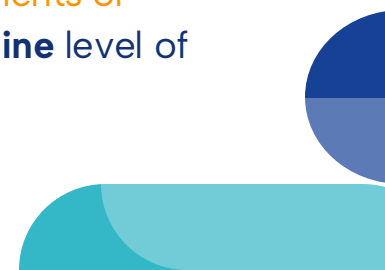
# Random intercepts

**Random intercepts** allow you to model how **level 2 (or higher) units deviate** from the population average.

For example, if examining the effect of pay scale on an employee's job satisfaction, you may have the following data structure:

Level 1: employee  
Level 2: department  
(Level 3: workplace)

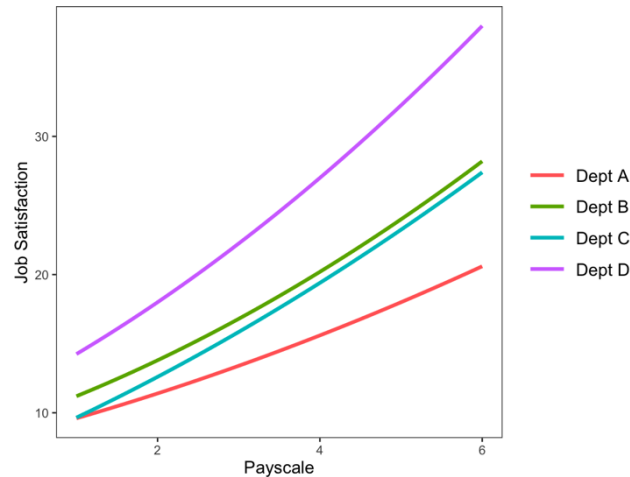
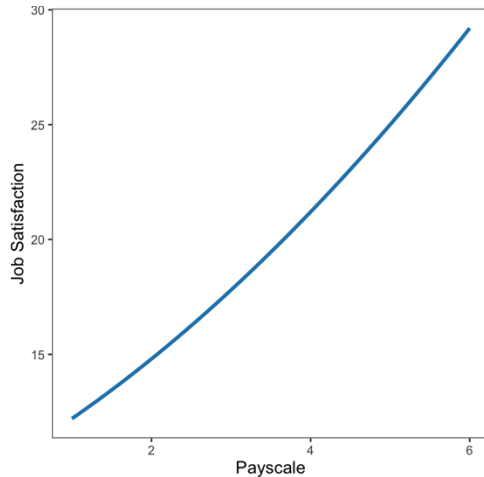
Random intercepts account for the fact that **employees in different departments or workplaces may have different average job satisfaction** — **shifting the baseline** level of the outcome for each group.



# Random intercepts

**Random intercepts** allow you to model how **level 2 (or higher) units deviate** from the population average.

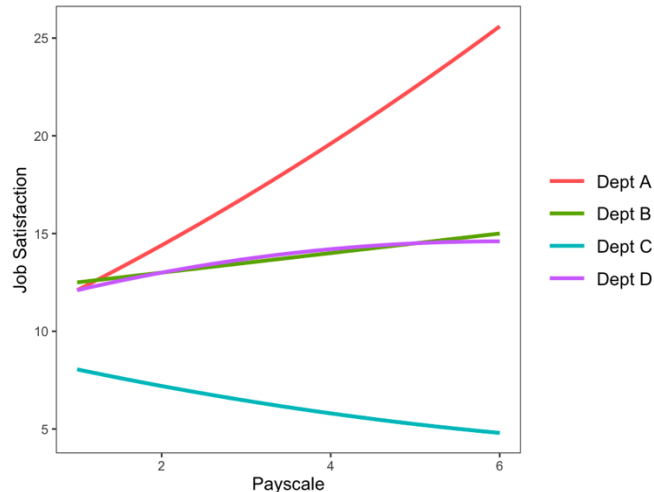
Random intercepts account for the fact that **employees in different departments or workplaces may have different average job satisfaction** — **shifting the baseline** level of the outcome for each group.



# Random slopes

**Random slopes** allow you to model how the effect of a predictor varies across higher-level groups (e.g., departments, classrooms).

In the same example as before, this could mean that in some departments, higher pay scales might not necessarily be associated with higher job satisfaction.





# Multi-Level Models in R

Syntax for specifying mixed effects models is similar to specifying a linear regression model, but uses the `lmer()` function for linear mixed effects regression and `glmer()` function for generalized mixed effects regression.

Random intercepts are modeled using the syntax `( 1 | random_intercept)`.

Random slopes are modeled using the syntax `( random_slope | random_intercept)`.

Syntax	Description
<code>outcome ~ fixed_1 + fixed_2 + ...</code>	Fixed effects model.
<code>outcome ~ fixed_1 + fixed_2 + (1   random_i)</code>	Random effects model with a random intercept.
<code>outcome ~ fixed_1 + fixed_2 + (random_s   random_i)</code>	Random effects model with a random intercept and a random slope in a nested design.
<code>outcome ~ fixed_1 + fixed_2 + (1   random_i1) + ( 1   random_i2)</code>	Random effects model with random intercepts in a crossed design.

# Job Satisfaction Survey

Job satisfaction survey of 399 university employees with the following data:

- Pay scale
- Department
- Job satisfaction ratings
- Department ratings

<https://uoepsy.github.io/data/>

Variable	Description	Class/type
NSSrating	National student survey rating of the university department	Continuous
dept	Department	Categorical
payscale	Payscale ranging from 5 – 10	Ordinal/categorical
jobsat	Job satisfaction rating	Continuous
jobsat_binary	Job satisfaction (yes/no)	Binary



# Thanks!

Do you have any questions?

shireen.parimoo@sickkids.ca



CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#), and infographics & images by [Freepik](#)