

# Package ‘moderndive’

June 30, 2024

**Type** Package

**Title** Tidyverse-Friendly Introductory Linear Regression

**Version** 0.6.1

**Maintainer** Albert Y. Kim <albert.y.s.kim@gmail.com>

**Description** Datasets and wrapper functions for tidyverse-friendly introductory linear regression, used in “Statistical Inference via Data Science: A ModernDive into R and the Tidyverse” available at <<https://moderndive.com/>>.

**Depends** R (>= 3.4.0)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**URL** <https://moderndive.github.io/moderndive/>,  
<https://github.com/moderndive/moderndive/>

**BugReports** <https://github.com/moderndive/moderndive/issues>

**Imports** magrittr, dplyr, ggplot2, tibble, janitor, broom (>= 0.4.3),  
formula.tools, stringr, knitr, infer, rlang (>= 0.2.0), glue

**RoxygenNote** 7.3.2

**Suggests** testthat, covr, rmarkdown, vdiff, openintro, patchwork,  
viridis, readr, nycflights13, nycflights23

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Albert Y. Kim [aut, cre] (<<https://orcid.org/0000-0001-7824-306X>>),  
Chester Ismay [aut] (<<https://orcid.org/0000-0003-2820-2547>>),  
Andrew Bray [ctb] (<<https://orcid.org/0000-0002-4037-7414>>),  
Delaney Moran [ctb],  
Evgeni Chasnovski [ctb] (<<https://orcid.org/0000-0002-1617-4019>>),  
Will Hopper [ctb] (<<https://orcid.org/0000-0002-7848-1946>>),  
Benjamin S. Baumer [ctb] (<<https://orcid.org/0000-0002-3279-0516>>),  
Marium Tapal [ctb] (<<https://orcid.org/0000-0001-5093-6462>>),  
Wayne Ndlovu [ctb],

Catherine Peppers [ctb],  
 Annah Mutaya [ctb],  
 Anushree Goswami [ctb],  
 Ziyue Yang [ctb] (<<https://orcid.org/0000-0002-9299-8327>>),  
 Clara Li [ctb] (<<https://orcid.org/0000-0003-2456-0849>>),  
 Caroline McKenna [ctb],  
 Catherine Park [ctb] (<<https://orcid.org/0000-0002-8273-9620>>),  
 Abbie Benfield [ctb],  
 Georgia Gans [ctb],  
 Kacey Jean-Jacques [ctb],  
 Swaha Bhattacharya [ctb],  
 Vivian Almaraz [ctb],  
 Elle Jo Whalen [ctb],  
 Jacqueline Chen [ctb],  
 Michelle Flesaker [ctb],  
 Irene Foster [ctb],  
 Aushanae Haller [ctb],  
 Benjamin Bruncati [ctb] (<<https://orcid.org/0000-0001-8545-5984>>),  
 Quinn White [ctb] (<<https://orcid.org/0000-0001-5399-0237>>),  
 Tianshu Zhang [ctb] (<<https://orcid.org/0000-0002-3004-4472>>),  
 Katelyn Diaz [ctb] (<<https://orcid.org/0000-0001-6108-1682>>),  
 Rose Porta [ctb],  
 Renee Wu [ctb],  
 Arris Moise [ctb],  
 Kate Phan [ctb],  
 Grace Hartley [ctb],  
 Silas Weden [ctb],  
 Emma Vejck [ctb],  
 Nikki Schuldt [ctb],  
 Tess Goldmann [ctb],  
 Hongtong Lin [ctb],  
 Alejandra Munoz [ctb],  
 Elina Gordon-Halpern [ctb],  
 Haley Schmidt [ctb] (<<https://orcid.org/0000-0002-6184-2266>>)

**Repository** CRAN

**Date/Publication** 2024-06-30 07:30:02 UTC

## Contents

alaska_flights . . . . .	3
almonds_bowl . . . . .	4
almonds_sample_100 . . . . .	5
amazon_books . . . . .	5
avocados . . . . .	6
babies . . . . .	6
bowl . . . . .	8
bowl_samples . . . . .	8

bowl_sample_1 . . . . .	9
coffee_ratings . . . . .	9
DD_vs_SB . . . . .	11
early_january_2023_weather . . . . .	11
early_january_weather . . . . .	12
envoy_flights . . . . .	13
evals . . . . .	14
ev_charging . . . . .	15
geom_categorical_model . . . . .	16
geom_parallel_slopes . . . . .	18
get_correlation . . . . .	21
get_regression_points . . . . .	22
get_regression_summaries . . . . .	23
get_regression_table . . . . .	24
gg_parallel_slopes . . . . .	25
house_prices . . . . .	27
ipf_lifts . . . . .	28
mario_kart_auction . . . . .	29
mass_traffic_2020 . . . . .	30
MA_schools . . . . .	30
ma_traffic_2020_vs_2019 . . . . .	31
moderndive . . . . .	32
movies_sample . . . . .	34
mythbusters_yawn . . . . .	34
orig_pennies_sample . . . . .	35
pennies . . . . .	35
pennies_resamples . . . . .	36
pennies_sample . . . . .	36
promotions . . . . .	37
promotions_shuffled . . . . .	37
saratoga_houses . . . . .	38
tactile_prop_red . . . . .	39
<b>Index</b>	<b>40</b>

---

alaska_flights	<i>Alaska flights data</i>
----------------	----------------------------

---

### Description

On-time data for all Alaska Airlines flights that departed NYC (i.e. JFK, LGA or EWR) in 2013. This is a subset of the `flights` data frame from `nycflights13`.

### Usage

```
alaska_flights
```

**Format**

A data frame of 714 rows representing Alaska Airlines flights and 19 variables

**year, month, day** Date of departure.

**dep\_time, arr\_time** Actual departure and arrival times (format HHMM or HMM), local tz.

**sched\_dep\_time, sched\_arr\_time** Scheduled departure and arrival times (format HHMM or HMM), local tz.

**dep\_delay, arr\_delay** Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.

**carrier** Two letter carrier abbreviation. See [nycflights13::airlines](#) to get name.

**flight** Flight number.

**tailnum** Plane tail number. See [nycflights13::planes](#) for additional metadata.

**origin, dest** Origin and destination. See [nycflights13::airports](#) for additional metadata.

**air\_time** Amount of time spent in the air, in minutes.

**distance** Distance between airports, in miles.

**hour, minute** Time of scheduled departure broken into hour and minutes.

**time\_hour** Scheduled date and hour of the flight as a POSIXct date. Along with origin, can be used to join flights data to [nycflights13::weather](#) data.

**Source**

RITA, Bureau of transportation statistics

**See Also**

[nycflights13::flights](#).

---

almonds\_bowl

*Chocolate-covered almonds data*

---

**Description**

5000 chocolate-covered almonds selected from a large batch, weighed in grams.

**Usage**

almonds\_bowl

**Format**

A data frame with 5000 observations on the following 2 variables

**ID** Identification value for a given chocolate-covered almond

**weight** Weight of the chocolate-covered almond in grams (to the nearest tenth)

---

almonds_sample_100	<i>Chocolate-covered almonds data sample</i>
--------------------	--

---

**Description**

A sample of 100 chocolate-covered almonds, weighed in grams.

**Usage**

almonds\_sample\_100

**Format**

A data frame with 100 observations on the following 2 variables

**ID** Identification value for a given chocolate-covered almond

**weight** Weight of the chocolate-covered almond in grams (to the nearest tenth)

---

amazon_books	<i>Sample of Amazon books</i>
--------------	-------------------------------

---

**Description**

A random sample of 325 books from Amazon.com.

**Usage**

amazon\_books

**Format**

A data frame of 325 rows representing books listed on Amazon and 13 variables.

**title** Book title

**author** Author who wrote book

**list\_price** recommended retail price of book

**amazon\_price** lowest price of book shown on Amazon

**hard\_paper** book is either hardcover or paperback

**num\_pages** number of pages in book

**publisher** Company that issues the book for sale

**pub\_year** Year the book was published

**isbn\_10** 10-character ISBN number

**height, width, thick, weight\_oz** height, width, weight and thickness of the book

**Source**

The Data and Story Library (DASL) <https://dasl.datadescription.com/datafile/amazon-books>

---

 avocados

*Avocado Prices by US Region*


---

**Description**

Gathered from <https://docs.google.com/spreadsheets/d/1cNuj9V-9Xe8fqV3DQRhvsXJhER3zTk01dSsQ1Q0j96g/edit#gid=1419070688>

**Usage**

avocados

**Format**

A data frame of 54 regions over 3 years of weekly results

**date** Week of Data Recording

**average\_price** Average Price of Avocado

**total\_volume** Total Amount of Avocados

**small\_hass\_sold** Amount of Small Haas Avocados Sold

**large\_hass\_sold** Amount of Large Haas Avocados Sold

**xlarge\_hass\_sold** Amount of Extra Large Haas Avocados Sold

**total\_bags** Total Amount of Bags of Avocados

**small\_bags** Total Amount of Bags of Small Haas Avocados

**large\_bags** Total Amount of Bags of Large Haas Avocados

**x\_large\_bags** Total Amount of Bags of Extra Large Haas Avocados

**type** Type of Sale

**year** Year of Sale

**region** Region Where Sale Took Place

---

 babies

*Data on maternal smoking and infant health*


---

**Description**

Data on maternal smoking and infant health

**Usage**

babies

**Format**

A data frame of 1236 rows of individual mothers.

**id** Identification number

**plurality** Marked 5 for single fetus, otherwise number of fetuses

**outcome** Marked 1 for live birth that survived at least 28 days

**date** Birth date where 1096 is January 1st, 1961

**birthday** Birth date in mm-dd-yyyy format

**gestation** Length of gestation in days, marked 999 if unknown

**sex** Infant's sex, where 1 is male, 2 is female, and 9 is unknown

**wt** Birth weight in ounces, marked 999 if unknown

**parity** Total number of previous pregnancies including fetal deaths and stillbirths, marked 99 if unknown

**race** Mother's race where 0-5 is white, 6 is Mexican, 7 is Black, 8 is Asian, 9 is mixed, and 99 is unknown

**age** Mother's age in years at termination of pregnancy, 99=unknown

**ed** Mother's education 0= less than 8th grade, 1 = 8th -12th grade - did not graduate, 2= HS graduate-no other schooling , 3= HS+trade, 4=HS+some college 5= College graduate, 6&7 Trade school HS unclear, 9=unknown

**ht** Mother's height in inches to the last completed inch, 99=unknown

**wt\_1** Mother prepregnancy wt in pounds, 999=unknown

**drace** Father's race, coding same as mother's race

**dage** Father's age, coding same as mother's age

**ded** Father's education, coding same as mother's education

**dht** Father's height, coding same as for mother's height

**dwt** Father's weight coding same as for mother's weight

**marital** 0= legally separated, 1=married, 2= divorced, 3=widowed, 5=never married

**inc** Family yearly income in \$2500 increments 0 = under 2500, 1=2500-4999, ..., 8= 12,500-14,999, 9=15000+, 98=unknown, 99=not asked

**smoke** Does mother smoke? 0=never, 1= smokes now, 2=until current pregnancy, 3=once did, not now, 9=unknown

**time** If mother quit, how long ago? 0=never smoked, 1=still smokes, 2=during current preg, 3=within 1 yr, 4= 1 to 2 years ago, 5= 2 to 3 yr ago, 6= 3 to 4 yrs ago, 7=5 to 9yrs ago, 8=10+yrs ago, 9=quit and don't know, 98=unknown, 99=not asked

**number** Number of cigs smoked per day for past and current smokers 0=never, 1=1-4, 2=5-9, 3=10-14, 4=15-19, 5=20-29, 6=30-39, 7=40-60, 8=60+, 9=smoke but don't know, 98=unknown, 99=not asked

**Source**

Data on maternal smoking and infant health from <https://www.stat.berkeley.edu/~statlabs/labs.html>

---

bowl	<i>A sampling bowl of red and white balls</i>
------	---

---

**Description**

A sampling bowl used as the population in a simulated sampling exercise. Also known as the urn sampling framework [https://en.wikipedia.org/wiki/Urn\\_problem](https://en.wikipedia.org/wiki/Urn_problem).

**Usage**

bowl

**Format**

A data frame 2400 rows representing different balls in the bowl, of which 900 are red and 1500 are white.

**ball\_ID** ID variable used to denote all balls. Note this value is not marked on the balls themselves  
**color** color of ball: red or white

---

bowl_samples	<i>Sampling from a bowl of balls</i>
--------------	--------------------------------------

---

**Description**

Counting the number of red balls in 10 samples of size  $n = 50$  balls from [https://github.com/moderndive/moderndive/blob/master/data-raw/sampling\\_bowl.jpeg](https://github.com/moderndive/moderndive/blob/master/data-raw/sampling_bowl.jpeg)

**Usage**

bowl\_samples

**Format**

A data frame 10 rows representing different groups of students' samples of size  $n = 50$  and 5 variables

**group** Group name  
**red** Number of red balls sampled  
**white** Number of white balls sampled  
**green** Number of green balls sampled  
**n** Total number of balls samples

**See Also**

[bowl\(\)](#)



---

bowl_sample_1	<i>Tactile sample of size 50 from a bowl of balls</i>
---------------	---

---

**Description**

A single tactile sample of size  $n = 50$  balls from [https://github.com/moderndive/moderndive/blob/master/data-raw/sampling\\_bowl.jpeg](https://github.com/moderndive/moderndive/blob/master/data-raw/sampling_bowl.jpeg)

**Usage**

```
bowl_sample_1
```

**Format**

A data frame of 50 rows representing different balls and 1 variable.

**color** Color of ball sampled

**See Also**

[bowl\(\)](#)

---

coffee_ratings	<i>Data from the Coffee Quality Institute's review pages in January 2018</i>
----------------	--

---

**Description**

1,340 digitized reviews on coffee samples from <https://database.coffeeinstitute.org/>.

**Usage**

```
coffee_ratings
```

**Format**

A data frame of 1,340 rows representing each sample of coffee.

**total\_cup\_points** Number of points in final rating (scale of 0-100)

**species** Species of coffee bean plant (Arabica or Robusta)

**owner** Owner of coffee plant farm

**country\_of\_origin** Coffee bean's country of origin

**farm\_name** Name of coffee plant farm

**lot\_number** Lot number for tested coffee beans

**mill** Name of coffee bean's processing facility

**ico\_number** International Coffee Organization number  
**company** Name of coffee bean's company  
**altitude** Altitude at which coffee plants were grown  
**region** Region where coffee plants were grown  
**producer** Name of coffee bean roaster  
**number\_of\_bags** Number of tested bags  
**bag\_weight** Tested bag weight  
**in\_country\_partner** Partner for the country  
**harvest\_year** Year the coffee beans were harvested  
**grading\_date** Day the coffee beans were graded  
**owner\_1** Owner of the coffee beans  
**variety** Variety of the coffee beans  
**processing\_method** Method used for processing the coffee beans  
**aroma** Coffee aroma rating  
**flavor** Coffee flavor rating  
**aftertaste** Coffee aftertaste rating  
**acidity** Coffee acidity rating  
**body** Coffee body rating  
**balance** Coffee balance rating  
**uniformity** Coffee uniformity rating  
**clean\_cup** Cup cleanliness rating  
**sweetness** Coffee sweetness rating  
**cupper\_points** Cupper Points, an overall rating for the coffee  
**moisture** Coffee moisture content  
**category\_one\_defects** Number of category one defects for the coffee beans  
**quakers** Number of coffee beans that don't dark brown when roasted  
**color** Color of the coffee beans  
**category\_two\_defects** Number of category two defects for the coffee beans  
**expiration** Expiration date of the coffee beans  
**certification\_body** Entity/Institute that certified the coffee beans  
**certification\_address** Body address of certification for coffee beans  
**certification\_contact** Certification contact for coffee beans  
**unit\_of\_measurement** Unit of measurement for altitude  
**altitude\_low\_meters** Lower altitude level coffee beans grow at  
**altitude\_high\_meters** Higher altitude level coffee beans grow at  
**altitude\_mean\_meters** Average altitude level coffee beans grow at

#### Source

Coffee Quality Institute. Access cleaned data available at <https://github.com/jldbc/coffee-quality-database>

---

`DD_vs_SB`*Dunkin Donuts vs Starbucks*

---

**Description**

Number of Dunkin Donuts & Starbucks, median income, and population in 1024 census tracts in eastern Massachusetts in 2016.

**Usage**`DD_vs_SB`**Format**

A data frame of 1024 rows representing census tracts and 6 variables

**county** County where census tract is located. Either Bristol, Essex, Middlesex, Norfolk, Plymouth, or Suffolk county

**FIPS** Federal Information Processing Standards code identifying census tract

**median\_income** Median income of census tract

**population** Population of census tract

**shop\_type** Coffee shop type: Dunkin Donuts or Starbucks

**shops** Number of shops

**Source**

US Census Bureau. Code used to scrape data available at <https://github.com/DelaneyMoran/FinalProject>

---

`early_january_2023_weather`*Early January hourly weather data for 2023*

---

**Description**

Hourly meteorological data for LGA, JFK and EWR for the month of January 2023. This is a subset of the weather data frame from `nycflights23`.

**Usage**`early_january_2023_weather`

**Format**

A data frame of 360 rows representing hourly measurements and 15 variables

**origin** Weather station. Named `origin` to facilitate merging with `nycflights23::flights` data.

**year, month, day, hour** Time of recording.

**temp, dewp** Temperature and dewpoint in F.

**humid** Relative humidity.

**wind\_dir, wind\_speed, wind\_gust** Wind direction (in degrees), speed and gust speed (in mph).

**precip** Precipitation, in inches.

**pressure** Sea level pressure in millibars.

**visib** Visibility in miles.

**time\_hour** Date and hour of the recording as a POSIXct date.

**Source**

ASOS download from Iowa Environmental Mesonet, <https://mesonet.agron.iastate.edu/request/download.phtml>.

**See Also**

[nycflights23::weather](#).

---

early\_january\_weather *Early January hourly weather data*

---

**Description**

Hourly meteorological data for LGA, JFK and EWR for the month of January 2013. This is a subset of the weather data frame from `nycflights13`.

**Usage**

```
early_january_weather
```

**Format**

A data frame of 358 rows representing hourly measurements and 15 variables

**origin** Weather station. Named `origin` to facilitate merging with `nycflights13::flights` data.

**year, month, day, hour** Time of recording.

**temp, dewp** Temperature and dewpoint in F.

**humid** Relative humidity.

**wind\_dir, wind\_speed, wind\_gust** Wind direction (in degrees), speed and gust speed (in mph).

**precip** Precipitation, in inches.

**pressure** Sea level pressure in millibars.

**visib** Visibility in miles.

**time\_hour** Date and hour of the recording as a POSIXct date.

**Source**

ASOS download from Iowa Environmental Mesonet, <https://mesonet.agron.iastate.edu/request/download.phtml>.

**See Also**

[nycflights13::weather](#).

---

envoy_flights	<i>Envoy Air flights data for 2023</i>
---------------	--

---

**Description**

On-time data for all Envoy Air flights that departed NYC (i.e. JFK, LGA or EWR) in 2023. This is a subset of the `flights` data frame from `nycflights23`.

**Usage**

```
envoy_flights
```

**Format**

A data frame of 357 rows representing Alaska Airlines flights and 19 variables

**year, month, day** Date of departure.

**dep\_time, arr\_time** Actual departure and arrival times (format HHMM or HMM), local tz.

**sched\_dep\_time, sched\_arr\_time** Scheduled departure and arrival times (format HHMM or HMM), local tz.

**dep\_delay, arr\_delay** Departure and arrival delays, in minutes. Negative times represent early departures/arrivals.

**carrier** Two letter carrier abbreviation. See [nycflights23::airlines](#) to get name.

**flight** Flight number.

**tailnum** Plane tail number. See [nycflights23::planes](#) for additional metadata.

**origin, dest** Origin and destination. See [nycflights23::airports](#) for additional metadata.

**air\_time** Amount of time spent in the air, in minutes.

**distance** Distance between airports, in miles.

**hour, minute** Time of scheduled departure broken into hour and minutes.

**time\_hour** Scheduled date and hour of the flight as a POSIXct date. Along with origin, can be used to join flights data to [nycflights23::weather](#) data.

**Source**

RITA, Bureau of transportation statistics

**See Also**

[nycflights23::flights](#).

---

evals

*Teaching evaluations at the UT Austin*

---

### Description

The data are gathered from end of semester student evaluations for a sample of 463 courses taught by 94 professors from the University of Texas at Austin. In addition, six students rate the professors' physical appearance. The result is a data frame where each row contains a different course and each column has information on either the course or the professor <https://www.openintro.org/data/index.php?data=evals>

### Usage

evals

### Format

A data frame with 463 observations corresponding to courses on the following 13 variables.

**ID** Identification variable for course.

**prof\_ID** Identification variable for professor. Many professors are included more than once in this dataset.

**score** Average professor evaluation score: (1) very unsatisfactory - (5) excellent.

**age** Age of professor.

**bty\_avg** Average beauty rating of professor.

**gender** Gender of professor (collected as a binary variable at the time of the study): female, male.

**ethnicity** Ethnicity of professor: not minority, minority.

**language** Language of school where professor received education: English or non-English.

**rank** Rank of professor: teaching, tenure track, tenured.

**pic\_outfit** Outfit of professor in picture: not formal, formal.

**pic\_color** Color of professor's picture: color, black & white.

**cls\_did\_eval** Number of students in class who completed evaluation.

**cls\_students** Total number of students in class.

**cls\_level** Class level: lower, upper.

### Source

Çetinkaya-Rundel M, Morgan KL, Stangl D. 2013. Looking Good on Course Evaluations. CHANCE 26(2).

### See Also

The data in `evals` is a slight modification of `openintro::evals()`.

---

 ev\_charging

*Electric vehicle charging sessions for a workplace charging program*


---

### Description

This dataset consists of information on 3,395 electric vehicle charging sessions across locations for a workplace charging program. The data contains information on multiple charging sessions from 85 electric vehicle drivers across 25 workplace locations, which are located at facilities of various types.

### Usage

ev\_charging

### Format

A data frame of 3,395 rows on 24 variables, where each row is an electric vehicle charging session.

**session\_id** Unique identifier specifying the electric vehicle charging session

**kwh\_total** Total energy used at the charging session, in kilowatt hours (kWh)

**dollars** Quantity of money paid for the charging session in U.S. dollars

**created** Date and time recorded at the beginning of the charging session

**ended** Date and time recorded at the end of the charging session

**start\_time** Hour of the day when the charging session began (1 through 24)

**end\_time** Hour of the day when the charging session ended (1 through 24)

**charge\_time\_hrs** Length of the charging session in hours

**weekday** First three characters of the name of the weekday when the charging session occurred

**platform** Digital platform the driver used to record the session (android, ios, web)

**distance** Distance from the charging location to the driver's home, expressed in miles NA if the driver did not report their address

**user\_id** Unique identifier for each driver

**station\_id** Unique identifier for each charging station

**location\_id** Unique identifier for each location owned by the company where charging stations were located

**manager\_vehicle** Binary variable that is 1 when the vehicle is a type commonly used by managers of the firm and 0 otherwise

**facility\_type** Categorical variable that represents the facility type:

- 1 = manufacturing
- 2 = office
- 3 = research and development
- 4 = other

**mon, tues, wed, thurs, fri, sat, sun** Binary variables; 1 if the charging session took place on that day, 0 otherwise

**reported\_zip** Binary variable; 1 if the driver did report their zip code, 0 if they did not

## Source

Harvard Dataverse [doi:10.7910/DVN/NFPQLW](https://doi.org/10.7910/DVN/NFPQLW). Note data is released under a CC0: Public Domain license.

---

geom\_categorical\_model

*Regression model with one categorical explanatory/predictor variable*

---

## Description

`geom_categorical_model()` fits a regression model using the categorical x axis as the explanatory variable, and visualizes the model's fitted values as piece-wise horizontal line segments. Confidence interval bands can be included in the visualization of the model. Like `geom_parallel_slopes()`, this function has the same nature as `geom_smooth()` from the `{ggplot2}` package, but provides functionality that `geom_smooth()` currently doesn't have. When using a categorical predictor variable, the intercept corresponds to the mean for the baseline group, while coefficients for the non-baseline groups are offsets from this baseline. Thus in the visualization the baseline for comparison group's median is marked with a solid line, whereas all offset groups' medians are marked with dashed lines.

## Usage

```
geom_categorical_model(  
  mapping = NULL,  
  data = NULL,  
  position = "identity",  
  ...,  
  se = TRUE,  
  level = 0.95,  
  na.rm = FALSE,  
  show.legend = NA,  
  inherit.aes = TRUE  
)
```

## Arguments

mapping	Set of aesthetic mappings created by <code>aes()</code> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	The data to be displayed in this layer. There are three options: If <code>NULL</code> , the default, the data is inherited from the plot data as specified in the call to <code>ggplot()</code> . A <code>data.frame</code> , or other object, will override the plot data. All objects will be fortified to produce a data frame. See <code>fortify()</code> for which variables will be created.



	A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code> , and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code> ).
position	<p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:</p> <ul style="list-style-type: none"> <li>• The result of calling a position function, such as <code>position_jitter()</code>. This method allows for passing extra arguments to the position.</li> <li>• A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <code>position_jitter()</code>, give the position as "jitter".</li> <li>• For more information and other ways to specify the position, see the <a href="#">layer position</a> documentation.</li> </ul>
...	<p>Other arguments passed on to <code>layer()</code>'s <code>params</code> argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the position argument, or aesthetics that are required can <i>not</i> be passed through ... Unknown arguments that are not part of the 4 categories below are ignored.</p> <ul style="list-style-type: none"> <li>• Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, <code>colour = "red"</code> or <code>linewidth = 3</code>. The geom's documentation has an <b>Aesthetics</b> section that lists the available options. The 'required' aesthetics cannot be passed on to the <code>params</code>. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.</li> <li>• When constructing a layer using a <code>stat_*()</code> function, the ... argument can be used to pass on parameters to the geom part of the layer. An example of this is <code>stat_density(geom = "area", outline.type = "both")</code>. The geom's documentation lists which parameters it can accept.</li> <li>• Inversely, when constructing a layer using a <code>geom_*()</code> function, the ... argument can be used to pass on parameters to the stat part of the layer. An example of this is <code>geom_area(stat = "density", adjust = 0.5)</code>. The stat's documentation lists which parameters it can accept.</li> <li>• The <code>key_glyph</code> argument of <code>layer()</code> may also be passed on through ... This can be one of the functions described as <a href="#">key glyphs</a>, to change the display of the layer in the legend.</li> </ul>
se	Display confidence interval around model lines? TRUE by default.
level	Level of confidence interval to use (0.95 by default).
na.rm	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
show.legend	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
inherit.aes	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .

**See Also**

[geom\\_parallel\\_slopes\(\)](#)

**Examples**

```
library(dplyr)
library(ggplot2)

p <- ggplot(mpg, aes(x = drv, y = hwy)) +
  geom_point() +
  geom_categorical_model()
p

# In the above visualization, the solid line corresponds to the mean of 19.2
# for the baseline group "4", whereas the dashed lines correspond to the
# means of 28.19 and 21.02 for the non-baseline groups "f" and "r" respectively.
# In the corresponding regression table however the coefficients for "f" and "r"
# are presented as offsets from the mean for "4":
model <- lm(hwy ~ drv, data = mpg)
get_regression_table(model)

# You can use different colors for each categorical level
p %+% aes(color = drv)

# But mapping the color aesthetic doesn't change the model that is fit
p %+% aes(color = class)
```

---

geom\_parallel\_slopes *Parallel slopes regression model*

---

**Description**

`geom_parallel_slopes()` fits parallel slopes model and adds its line output(s) to a ggplot object. Basically, it fits a unified model with intercepts varying between groups (which should be supplied as standard {ggplot2} grouping aesthetics: `group`, `color`, `fill`, etc.). This function has the same nature as `geom_smooth()` from {ggplot2} package, but provides functionality that `geom_smooth()` currently doesn't have.

**Usage**

```
geom_parallel_slopes(
  mapping = NULL,
  data = NULL,
  position = "identity",
  ...,
  se = TRUE,
  formula = y ~ x,
  n = 100,
```

```

    fullrange = FALSE,
    level = 0.95,
    na.rm = FALSE,
    show.legend = NA,
    inherit.aes = TRUE
  )

```

## Arguments

mapping	Set of aesthetic mappings created by <a href="#">aes()</a> . If specified and <code>inherit.aes = TRUE</code> (the default), it is combined with the default mapping at the top level of the plot. You must supply mapping if there is no plot mapping.
data	<p>The data to be displayed in this layer. There are three options:</p> <p>If <code>NULL</code>, the default, the data is inherited from the plot data as specified in the call to <a href="#">ggplot()</a>.</p> <p>A <code>data.frame</code>, or other object, will override the plot data. All objects will be fortified to produce a data frame. See <a href="#">fortify()</a> for which variables will be created.</p> <p>A function will be called with a single argument, the plot data. The return value must be a <code>data.frame</code>, and will be used as the layer data. A function can be created from a formula (e.g. <code>~ head(.x, 10)</code>).</p>
position	<p>A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The position argument accepts the following:</p> <ul style="list-style-type: none"> <li>• The result of calling a position function, such as <a href="#">position_jitter()</a>. This method allows for passing extra arguments to the position.</li> <li>• A string naming the position adjustment. To give the position as a string, strip the function name of the <code>position_</code> prefix. For example, to use <a href="#">position_jitter()</a>, give the position as "jitter".</li> <li>• For more information and other ways to specify the position, see the <a href="#">layer position</a> documentation.</li> </ul>
...	<p>Other arguments passed on to <a href="#">layer()</a>'s <code>params</code> argument. These arguments broadly fall into one of 4 categories below. Notably, further arguments to the position argument, or aesthetics that are required can <i>not</i> be passed through ... Unknown arguments that are not part of the 4 categories below are ignored.</p> <ul style="list-style-type: none"> <li>• Static aesthetics that are not mapped to a scale, but are at a fixed value and apply to the layer as a whole. For example, <code>colour = "red"</code> or <code>linewidth = 3</code>. The geom's documentation has an <b>Aesthetics</b> section that lists the available options. The 'required' aesthetics cannot be passed on to the <code>params</code>. Please note that while passing unmapped aesthetics as vectors is technically possible, the order and required length is not guaranteed to be parallel to the input data.</li> <li>• When constructing a layer using a <code>stat_*()</code> function, the ... argument can be used to pass on parameters to the geom part of the layer. An example of this is <code>stat_density(geom = "area", outline.type = "both")</code>. The geom's documentation lists which parameters it can accept.</li> </ul>

- Inversely, when constructing a layer using a `geom_*()` function, the `...` argument can be used to pass on parameters to the `stat` part of the layer. An example of this is `geom_area(stat = "density", adjust = 0.5)`. The `stat`'s documentation lists which parameters it can accept.
- The `key_glyph` argument of `layer()` may also be passed on through `...`. This can be one of the functions described as [key glyphs](#), to change the display of the layer in the legend.

<code>se</code>	Display confidence interval around model lines? TRUE by default.
<code>formula</code>	Formula to use per group in parallel slopes model. Basic linear $y \sim x$ by default.
<code>n</code>	Number of points per group at which to evaluate model.
<code>fullrange</code>	If TRUE, the smoothing line gets expanded to the range of the plot, potentially beyond the data. This does not extend the line into any additional padding created by expansion.
<code>level</code>	Level of confidence interval to use (0.95 by default).
<code>na.rm</code>	If FALSE, the default, missing values are removed with a warning. If TRUE, missing values are silently removed.
<code>show.legend</code>	logical. Should this layer be included in the legends? NA, the default, includes if any aesthetics are mapped. FALSE never includes, and TRUE always includes. It can also be a named logical vector to finely select the aesthetics to display.
<code>inherit.aes</code>	If FALSE, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. <code>borders()</code> .

### See Also

[geom\\_categorical\\_model\(\)](#)

### Examples

```
library(dplyr)
library(ggplot2)

ggplot(evals, aes(x = age, y = score, color = ethnicity)) +
  geom_point() +
  geom_parallel_slopes(se = FALSE)

# Basic usage
ggplot(evals, aes(x = age, y = score, color = ethnicity)) +
  geom_point() +
  geom_parallel_slopes()
ggplot(evals, aes(x = age, y = score, color = ethnicity)) +
  geom_point() +
  geom_parallel_slopes(se = FALSE)

# Supply custom aesthetics
ggplot(evals, aes(x = age, y = score, color = ethnicity)) +
  geom_point() +
  geom_parallel_slopes(se = FALSE, size = 4)
```

```
# Fit non-linear model
example_df <- house_prices %>%
  slice(1:1000) %>%
  mutate(
    log10_price = log10(price),
    log10_size = log10(sqft_living)
  )
ggplot(example_df, aes(x = log10_size, y = log10_price, color = condition)) +
  geom_point(alpha = 0.1) +
  geom_parallel_slopes(formula = y ~ poly(x, 2))

# Different grouping
ggplot(example_df, aes(x = log10_size, y = log10_price)) +
  geom_point(alpha = 0.1) +
  geom_parallel_slopes(aes(fill = condition))
```

---

get_correlation	<i>Get correlation value in a tidy way</i>
-----------------	--

---

## Description

Determine the Pearson correlation coefficient between two variables in a data frame using pipeable and formula-friendly syntax

## Usage

```
get_correlation(data, formula, na.rm = FALSE, ...)
```

## Arguments

data	a data frame object
formula	a formula with the response variable name on the left and the explanatory variable name on the right
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
...	further arguments passed to <a href="#">stats::cor()</a>

## Value

A 1x1 data frame storing the correlation value

## Examples

```
library(moderndive)

# Compute correlation between mpg and cyl:
mtcars %>%
```

```

get_correlation(formula = mpg ~ cyl)

# Group by one variable:
library(dplyr)
mtcars %>%
  group_by(am) %>%
  get_correlation(formula = mpg ~ cyl)

# Group by two variables:
mtcars %>%
  group_by(am, gear) %>%
  get_correlation(formula = mpg ~ cyl)

```

---

get\_regression\_points *Get regression points*

---

### Description

Output information on each point/observation used in an `lm()` regression in "tidy" format. This function is a wrapper function for `broom::augment()` and renames the variables to have more intuitive names.

### Usage

```

get_regression_points(
  model,
  digits = 3,
  print = FALSE,
  newdata = NULL,
  ID = NULL
)

```

### Arguments

<code>model</code>	an <code>lm()</code> model object
<code>digits</code>	number of digits precision in output table
<code>print</code>	If TRUE, return in print format suitable for R Markdown
<code>newdata</code>	A new data frame of points/observations to apply <code>model</code> to obtain new fitted values and/or predicted values $\hat{y}$ . Note the format of <code>newdata</code> must match the format of the original data used to fit <code>model</code> .
<code>ID</code>	A string indicating which variable in either the original data used to fit <code>model</code> or <code>newdata</code> should be used as an identification variable to distinguish the observational units in each row. This variable will be the left-most variable in the output data frame. If <code>ID</code> is unspecified, a column <code>ID</code> with values 1 through the number of rows is returned as the identification variable.

**Value**

A tibble-formatted regression table of outcome/response variable, all explanatory/predictor variables, the fitted/predicted value, and residual.

**See Also**

[augment\(\)](#), [get\\_regression\\_table\(\)](#), [get\\_regression\\_summaries\(\)](#)

**Examples**

```
library(dplyr)
library(tibble)

# Convert rownames to column
mtcars <- mtcars %>%
  rownames_to_column(var = "automobile")

# Fit lm() regression:
mpg_model <- lm(mpg ~ cyl, data = mtcars)

# Get information on all points in regression:
get_regression_points(mpg_model, ID = "automobile")

# Create training and test set based on mtcars:
training_set <- mtcars %>%
  sample_frac(0.5)
test_set <- mtcars %>%
  anti_join(training_set, by = "automobile")

# Fit model to training set:
mpg_model_train <- lm(mpg ~ cyl, data = training_set)

# Make predictions on test set:
get_regression_points(mpg_model_train, newdata = test_set, ID = "automobile")
```

---

get\_regression\_summaries

*Get regression summary values*

---

**Description**

Output scalar summary statistics for an `lm()` regression in "tidy" format. This function is a wrapper function for `broom::glance()`.

**Usage**

```
get_regression_summaries(model, digits = 3, print = FALSE)
```

**Arguments**

model	an <code>lm()</code> model object
digits	number of digits precision in output table
print	If TRUE, return in print format suitable for R Markdown

**Value**

A single-row tibble with regression summaries. Ex: `r_squared` and `mse`.

**See Also**

[glance\(\)](#), [get\\_regression\\_table\(\)](#), [get\\_regression\\_points\(\)](#)

**Examples**

```
library(moderndiver)

# Fit lm() regression:
mpg_model <- lm(mpg ~ cyl, data = mtcars)

# Get regression summaries:
get_regression_summaries(mpg_model)
```

---

`get_regression_table` *Get regression table*

---

**Description**

Output regression table for an `lm()` regression in "tidy" format. This function is a wrapper function for `broom::tidy()` and includes confidence intervals in the output table by default.

**Usage**

```
get_regression_table(
  model,
  conf.level = 0.95,
  digits = 3,
  print = FALSE,
  default_categorical_levels = FALSE
)
```



**Arguments**

model	an <code>lm()</code> model object
conf.level	The confidence level to use for the confidence interval if <code>conf.int = TRUE</code> . Must be strictly greater than 0 and less than 1. Defaults to 0.95, which corresponds to a 95 percent confidence interval.
digits	number of digits precision in output table
print	If TRUE, return in print format suitable for R Markdown
default_categorical_levels	If TRUE, do not change the non-baseline categorical variables in the term column. Otherwise non-baseline categorical variables will be displayed in the format "categorical_variable_name: level_name"

**Value**

A tibble-formatted regression table along with lower and upper end points of all confidence intervals for all parameters `lower_ci` and `upper_ci`; the confidence levels default to 95%

**See Also**

[tidy\(\)](#), [get\\_regression\\_points\(\)](#), [get\\_regression\\_summaries\(\)](#)

**Examples**

```
library(moderndiver)

# Fit lm() regression:
mpg_model <- lm(mpg ~ cyl, data = mtcars)

# Get regression table:
get_regression_table(mpg_model)

# Vary confidence level of confidence intervals
get_regression_table(mpg_model, conf.level = 0.99)
```

---

`gg_parallel_slopes`      *Plot parallel slopes model*

---

**Description**

NOTE: This function is deprecated; please use [geom\\_parallel\\_slopes\(\)](#) instead. Output a visualization of linear regression when you have one numerical and one categorical explanatory/predictor variable: a separate colored regression line for each level of the categorical variable

**Usage**

```
gg_parallel_slopes(y, num_x, cat_x, data, alpha = 1)
```

**Arguments**

<code>y</code>	Character string of outcome variable in data
<code>num_x</code>	Character string of numerical explanatory/predictor variable in data
<code>cat_x</code>	Character string of categorical explanatory/predictor variable in data
<code>data</code>	an optional data frame, list or environment (or object coercible by <code>as.data.frame</code> to a data frame) containing the variables in the model. If not found in <code>data</code> , the variables are taken from <code>environment(formula)</code> , typically the environment from which <code>lm</code> is called.
<code>alpha</code>	Transparency of points

**Value**

A `ggplot2::ggplot()` object.

**See Also**

[geom\\_parallel\\_slopes\(\)](#)

**Examples**

```
## Not run:
library(ggplot2)
library(dplyr)
library(moderndiver)

# log10() transformations
house_prices <- house_prices %>%
  mutate(
    log10_price = log10(price),
    log10_size = log10(sqft_living)
  )

# Output parallel slopes model plot:
gg_parallel_slopes(
  y = "log10_price", num_x = "log10_size", cat_x = "condition",
  data = house_prices, alpha = 0.1
) +
  labs(
    x = "log10 square feet living space", y = "log10 price in USD",
    title = "House prices in Seattle: Parallel slopes model"
  )

# Compare with interaction model plot:
ggplot(house_prices, aes(x = log10_size, y = log10_price, col = condition)) +
  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", se = FALSE, size = 1) +
  labs(
    x = "log10 square feet living space", y = "log10 price in USD",
    title = "House prices in Seattle: Interaction model"
  )
)
```

```
## End(Not run)
```

---

house_prices	<i>House Sales in King County, USA</i>
--------------	--

---

## Description

This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015. This dataset was obtained from Kaggle.com <https://www.kaggle.com/harlfoxem/housesalesprediction/data>

## Usage

```
house_prices
```

## Format

A data frame with 21613 observations on the following 21 variables.

**id** a notation for a house

**date** Date house was sold

**price** Price is prediction target

**bedrooms** Number of Bedrooms/House

**bathrooms** Number of bathrooms/bedrooms

**sqft\_living** square footage of the home

**sqft\_lot** square footage of the lot

**floors** Total floors (levels) in house

**waterfront** House which has a view to a waterfront

**view** Has been viewed

**condition** How good the condition is (Overall)

**grade** overall grade given to the housing unit, based on King County grading system

**sqft\_above** square footage of house apart from basement

**sqft\_basement** square footage of the basement

**yr\_built** Built Year

**yr\_renovated** Year when house was renovated

**zipcode** zip code

**lat** Latitude coordinate

**long** Longitude coordinate

**sqft\_living15** Living room area in 2015 (implies– some renovations) This might or might not have affected the lotsize area

**sqft\_lot15** lotSize area in 2015 (implies– some renovations)

**Source**

Kaggle <https://www.kaggle.com/harlfoxem/housesalesprediction>. Note data is released under a CC0: Public Domain license.

---

ipf_lifts	<i>International Power Lifting Results A subset of international powerlifting results.</i>
-----------	--

---

**Description**

International Power Lifting Results A subset of international powerlifting results.

**Usage**

ipf\_lifts

**Format**

A data frame with 41,152 entries, one entry for individual lifter

**name** Individual lifter name

**sex** Binary sex (M/F)

**event** The type of competition that the lifter entered

**equipment** The equipment category under which the lifts were performed

**age** The age of the lifter on the start date of the meet

**age\_class** The age class in which the filter falls

**division** division of competition

**bodyweight\_kg** The recorded bodyweight of the lifter at the time of competition, to two decimal places

**weight\_class\_kg** The weight class in which the lifter competed, to two decimal places

**best3squat\_kg** Maximum of the first three successful attempts for the lift

**best3bench\_kg** Maximum of the first three successful attempts for the lift

**best3deadlift\_kg** Maximum of the first three successful attempts for the lift

**place** The recorded place of the lifter in the given division at the end of the meet

**date** Date of the event

**federation** The federation that hosted the meet

**meet\_name** The name of the meet

**Source**

This data is a subset of the open dataset [Open Powerlifting](#)

---

mario\_kart\_auction      *Data from Mario Kart Ebay auctions*

---

### Description

Ebay auction data for the Nintendo Wii game Mario Kart.

### Usage

mario\_kart\_auction

### Format

A data frame of 143 auctions.

**id** Auction ID assigned by Ebay

**duration** Auction length in days

**n\_bids** Number of bids

**cond** Game condition, either new or used

**start\_pr** Price at the start of the auction

**ship\_pr** Shipping price

**total\_pr** Total price, equal to auction price plus shipping price

**ship\_sp** Shipping speed or method

**seller\_rate** Seller's rating on Ebay, equal to the number of positive ratings minus the number of negative ratings

**stock\_photo** Whether the auction photo was a stock photo or not, pictures used in many options were considered stock photos

**wheels** Number of Wii wheels included in the auction

**title** The title of the auctions

### Source

This data is from <https://www.openintro.org/data/index.php?data=mariokart>

---

mass_traffic_2020	<i>2020 road traffic volume and crash level date for 13 Massachusetts counties</i>
-------------------	--

---

**Description**

2020 road traffic volume and crash level date for 13 Massachusetts counties

**Usage**

mass\_traffic\_2020

**Format**

A data frame of 874 rows representing traffic data at the 874 sites

**site\_id** Site id

**county** County in which the site is located

**community** Community in which the site is located

**rural\_urban** Rural (R) or Urban (U)

**dir** Direction for traffic movement. Either 1-WAY, 2-WAY, EB (eastbound), RAMP or WB (west-bound)

**functional\_class** Classification of road. Either Arterial, Collector, Freeway & Expressway, Interstate or Local Road

**avg\_speed** Average traffic speed

**total\_volume** Number of vehicles recorded at each site in 2020

**crashes** Number of vehicle crashes at each site

**nonfatal\_injuries** Number of non-fatal injuries for all recorded vehicle crashes

**fatal\_injuries** Number of fatal injuries for all recorded vehicle crashes

---

MA_schools	<i>Massachusetts Public High Schools Data</i>
------------	---

---

**Description**

Data on Massachusetts public high schools in 2017

**Usage**

MA\_schools

**Format**

A data frame of 332 rows representing Massachusetts high schools and 4 variables

**school\_name** High school name.

**average\_sat\_math** Average SAT math score. Note 58 of the original 390 values of this variable were missing; these rows were dropped from consideration.

**perc\_disadvan** Percent of the student body that are considered economically disadvantaged.

**size** Size of school enrollment; small 13-341 students, medium 342-541 students, large 542-4264 students.

**Source**

The original source of the data are Massachusetts Department of Education reports [https://profiles.doe.mass.edu/state\\_report/](https://profiles.doe.mass.edu/state_report/), however the data was downloaded from Kaggle at <https://www.kaggle.com/ndalziel/massachusetts-public-schools-data>

---

ma\_traffic\_2020\_vs\_2019

*Massachusetts 2020 vs. 2019 Traffic Data Comparison*

---

**Description**

This dataset contains information about changes in speed, volume, and accidents of traffic between 2020 and 2019 by community and class of road in Massachusetts.

**Usage**

ma\_traffic\_2020\_vs\_2019

**Format**

A data frame of 264 rows each representing a different community in Massachusetts.

**community** City or Town

**functional\_class** Class or group the road belongs to

**change\_in\_speed** Average estimated Speed (mph)

**change\_in\_volume** Average traffic

**change\_in\_accidents** Average number of accidents

**Source**

<https://massdot-impact-crashes-vhb.opendata.arcgis.com/datasets/MassDOT::2020-vehicle-level-crash-c>  
[explore https://mhd.public.ms2soft.com/tcads/tsearch.asp?loc=Mhd&mod=](https://mhd.public.ms2soft.com/tcads/tsearch.asp?loc=Mhd&mod=)

---

moderndive

*moderndive - Tidyverse-Friendly Introductory Linear Regression*

---

## Description

Datasets and wrapper functions for tidyverse-friendly introductory linear regression, used in "Statistical Inference via Data Science: A ModernDive into R and the tidyverse" available at <https://moderndive.com/>.

## Author(s)

**Maintainer:** Albert Y. Kim <albert.y.s.kim@gmail.com> ([ORCID](#))

Authors:

- Chester Ismay <chester.ismay@gmail.com> ([ORCID](#))

Other contributors:

- Andrew Bray <bray@reed.edu> ([ORCID](#)) [contributor]
- Delaney Moran <delaneyemoran@gmail.com> [contributor]
- Evgeni Chasnovski <evgeni.chasnovski@gmail.com> ([ORCID](#)) [contributor]
- Will Hopper <wjhopper510@gmail.com> ([ORCID](#)) [contributor]
- Benjamin S. Baumer <ben.baumer@gmail.com> ([ORCID](#)) [contributor]
- Marium Tapal <mariumtapal@gmail.com> ([ORCID](#)) [contributor]
- Wayne Ndlovu <waynedndlovu5@gmail.com> [contributor]
- Catherine Peppers <cpeppers@smith.edu> [contributor]
- Annah Mutaya <annahmutaya18@gmail.com> [contributor]
- Anushree Goswami <anushreegoswami@gmail.com> [contributor]
- Ziyue Yang <zyang2k@gmail.com> ([ORCID](#)) [contributor]
- Clara Li <clarasepianli@gmail.com> ([ORCID](#)) [contributor]
- Caroline McKenna <carolinemckenna101@gmail.com> [contributor]
- Catherine Park <jcathyp@gmail.com> ([ORCID](#)) [contributor]
- Abbie Benfield <abbidabbers@gmail.com> [contributor]
- Georgia Gans <georgiagans@live.com> [contributor]
- Kacey Jean-Jacques <kjeanjacques@smith.edu> [contributor]
- Swaha Bhattacharya <itsswahabhattacharya@gmail.com> [contributor]
- Vivian Almaraz <viviansofia101@gmail.com> [contributor]
- Elle Jo Whalen <ewhalen@smith.edu> [contributor]
- Jacqueline Chen <jchen76@smith.edu> [contributor]
- Michelle Flesaker <mflesaker@smith.edu> [contributor]
- Irene Foster <ifoster25@smith.edu> [contributor]



- Aushanae Haller <aushanaehaller@gmail.com> [contributor]
- Benjamin Bruncati <kbruncati@smith.edu> (ORCID) [contributor]
- Quinn White <quinnarlise@gmail.com> (ORCID) [contributor]
- Tianshu Zhang <tzhang26@smith.edu> (ORCID) [contributor]
- Katelyn Diaz <katndiaz@gmail.com> (ORCID) [contributor]
- Rose Porta <rporta@smith.edu> [contributor]
- Renee Wu <rwu30@smith.edu> [contributor]
- Arris Moise <amoise@smith.edu> [contributor]
- Kate Phan <kphan@smith.edu> [contributor]
- Grace Hartley <grace.hartley@gmail.com> [contributor]
- Silas Weden <silasweden@gmail.com> [contributor]
- Emma Vejckic <evejckic@gmail.com> (ORCID) [contributor]
- Nikki Schuldt <nikkischuldt@gmail.com> [contributor]
- Tess Goldmann <tessgoldmann@aol.com> [contributor]
- Hongtong Lin <cccynthia1ht@gmail.com> [contributor]
- Alejandra Munoz <amunozgarcia@smith.edu> [contributor]
- Elina Gordon-Halpern <egordonhalpern@smith.edu> [contributor]
- Haley Schmidt <heschmidt00@gmail.com> (ORCID) [contributor]

### See Also

Useful links:

- <https://moderndive.github.io/moderndive/>
- <https://github.com/moderndive/moderndive/>
- Report bugs at <https://github.com/moderndive/moderndive/issues>

### Examples

```
library(moderndive)

# Fit regression model:
mpg_model <- lm(mpg ~ hp, data = mtcars)

# Regression tables:
get_regression_table(mpg_model)

# Information on each point in a regression:
get_regression_points(mpg_model)

# Regression summaries
get_regression_summaries(mpg_model)

# Plotting parallel slopes models
library(ggplot2)
```

```
ggplot(evals, aes(x = age, y = score, color = ethnicity)) +
  geom_point() +
  geom_parallel_slopes(se = FALSE)
```

---

movies_sample	<i>Random sample of 68 action and romance movies</i>
---------------	--

---

### Description

A random sample of 32 action movies and 36 romance movies from <https://www.imdb.com/> and their ratings.

### Usage

```
movies_sample
```

### Format

A data frame of 68 rows movies.

**title** Movie title

**year** Year released

**rating** IMDb rating out of 10 stars

**genre** Action or Romance

### See Also

This data was sampled from the movies data frame in the ggplot2movies package.

---

mythbusters_yawn	<i>Data from Mythbusters' study on contagiousness of yawning</i>
------------------	--

---

### Description

From a study on whether yawning is contagious <https://www.imdb.com/title/tt0768479/>. The data here was derived from the final proportions of yawns given in the show.

### Usage

```
mythbusters_yawn
```

### Format

A data frame of 50 rows representing each of the 50 participants in the study.

**subj** integer value corresponding to identifier variable of subject ID

**group** string of either "seed", participant was shown a yawner, or "control", participant was not shown a yawner

**yawn** string of either "yes", the participant yawned, or "no", the participant did not yawn

---

orig\_pennies\_sample     *A random sample of 40 pennies sampled from the pennies data frame*

---

**Description**

A dataset of 40 pennies to be treated as a random sample with `pennies()` acting as the population. Data on these pennies were recorded in 2011.

**Usage**

```
orig_pennies_sample
```

**Format**

A data frame of 40 rows representing 40 randomly sampled pennies from `pennies()` and 2 variables

**year** Year of minting

**age\_in\_2011** Age in 2011

**Source**

StatCrunch <https://www.statcrunch.com:443/app/index.html?dataid=301596>

**See Also**

`pennies()`

---

pennies     *A population of 800 pennies sampled in 2011*

---

**Description**

A dataset of 800 pennies to be treated as a sampling population. Data on these pennies were recorded in 2011.

**Usage**

```
pennies
```

**Format**

A data frame of 800 rows representing different pennies and 2 variables

**year** Year of minting

**age\_in\_2011** Age in 2011

**Source**

StatCrunch <https://www.statcrunch.com:443/app/index.html?dataid=301596>

---

pennies\_resamples      *Bootstrap resamples of a sample of 50 pennies*

---

### Description

35 bootstrap resamples with replacement of sample of 50 pennies contained in a 50 cent roll from Florence Bank on Friday February 1, 2019 in downtown Northampton, Massachusetts, USA <https://goo.gl/maps/AF88fpvVfm12>. The original sample of 50 pennies is available in `pennies_sample()`.

### Usage

```
pennies_resamples
```

### Format

A data frame of 1750 rows representing 35 students' bootstrap resamples of size 50 and 3 variables

**replicate** ID variable of replicate/resample number.

**name** Name of student

**year** Year on resampled penny

### See Also

[pennies\\_sample\(\)](#)

---

pennies\_sample      *A sample of 50 pennies*

---

### Description

A sample of 50 pennies contained in a 50 cent roll from Florence Bank on Friday February 1, 2019 in downtown Northampton, Massachusetts, USA <https://goo.gl/maps/AF88fpvVfm12>.

### Usage

```
pennies_sample
```

### Format

A data frame of 50 rows representing 50 sampled pennies and 2 variables

**ID** Variable used to uniquely identify each penny.

**year** Year of minting.

### Note

The original `pennies_sample` has been renamed `orig_pennies_sample()` as of `moderndive` v0.3.0.

---

promotions	<i>Bank manager recommendations based on (binary) gender</i>
------------	--

---

**Description**

Data from a 1970's study on whether gender influences hiring recommendations. Originally used in OpenIntro.org.

**Usage**

```
promotions
```

**Format**

A data frame with 48 observations on the following 3 variables.

**id** Identification variable used to distinguish rows.

**gender** gender (collected as a binary variable at the time of the study): a factor with two levels male and female

**decision** a factor with two levels: promoted and not

**Source**

Rosen B and Jerdee T. 1974. Influence of sex role stereotypes on personnel decisions. Journal of Applied Psychology 59(1):9-14.

**See Also**

The data in promotions is a slight modification of [openintro::gender\\_discrimination\(\)](#).

---

promotions_shuffled	<i>One permutation/shuffle of promotions</i>
---------------------	--

---

**Description**

Shuffled/permuted data from a 1970's study on whether gender influences hiring recommendations.

**Usage**

```
promotions_shuffled
```

**Format**

A data frame with 48 observations on the following 3 variables.

**id** Identification variable used to distinguish rows.

**gender** shuffled/permuted (binary) gender: a factor with two levels male and female

**decision** a factor with two levels: promoted and not

**See Also**

[promotions\(\)](#).

---

saratoga_houses	<i>House Prices and Properties in Saratoga, New York</i>
-----------------	--

---

**Description**

Random sample of 1057 houses taken from full Saratoga Housing Data (De Veaux)

**Usage**

saratoga\_houses

**Format**

A data frame with 1057 observations on the following 8 variables

**price** price (US dollars)

**living\_area** Living Area (square feet)

**bathrooms** Number of Bathroom (half bathrooms have no shower or tub)

**bedrooms** Number of Bedrooms

**fireplaces** Number of Fireplaces

**lot\_size** Size of Lot (acres)

**age** Age of House (years)

**fireplace** Whether the house has a Fireplace

**Source**

Gathered from <https://docs.google.com/spreadsheets/d/1AY5eECqNIggKpYF3kYzJQBIuuOdkic1FhbjAmY3Yc8E/edit#gid=622599674>

---

tactile_prop_red	<i>Tactile sampling from a tub of balls</i>
------------------	---

---

**Description**

Counting the number of red balls in 33 tactile samples of size  $n = 50$  balls from [https://github.com/moderndive/moderndive/blob/master/data-raw/sampling\\_bowl.jpeg](https://github.com/moderndive/moderndive/blob/master/data-raw/sampling_bowl.jpeg)

**Usage**

```
tactile_prop_red
```

**Format**

A data frame of 33 rows representing different groups of students' samples of size  $n = 50$  and 4 variables

**group** Group members

**replicate** Replicate number

**red\_balls** Number of red balls sampled out of 50

**prop\_red** Proportion red balls out of 50

**See Also**

[bowl\(\)](#)

# Index

## \* datasets

- alaska\_flights, 3
- almonds\_bowl, 4
- almonds\_sample\_100, 5
- amazon\_books, 5
- avocados, 6
- babies, 6
- bowl, 8
- bowl\_sample\_1, 9
- bowl\_samples, 8
- coffee\_ratings, 9
- DD\_vs\_SB, 11
- early\_january\_2023\_weather, 11
- early\_january\_weather, 12
- envoy\_flights, 13
- ev\_charging, 15
- evals, 14
- house\_prices, 27
- ipf\_lifts, 28
- MA\_schools, 30
- ma\_traffic\_2020\_vs\_2019, 31
- mario\_kart\_auction, 29
- mass\_traffic\_2020, 30
- movies\_sample, 34
- mythbusters\_yawn, 34
- orig\_pennies\_sample, 35
- pennies, 35
- pennies\_resamples, 36
- pennies\_sample, 36
- promotions, 37
- promotions\_shuffled, 37
- saratoga\_houses, 38
- tactile\_prop\_red, 39

augment(), 23

avocados, 6

babies, 6

borders(), 17, 20

bowl, 8

bowl(), 8, 9, 39

bowl\_sample\_1, 9

bowl\_samples, 8

coffee\_ratings, 9

DD\_vs\_SB, 11

early\_january\_2023\_weather, 11

early\_january\_weather, 12

envoy\_flights, 13

ev\_charging, 15

evals, 14

fortify(), 16, 19

geom\_categorical\_model, 16

geom\_categorical\_model(), 20

geom\_parallel\_slopes, 18

geom\_parallel\_slopes(), 16, 18, 25, 26

get\_correlation, 21

get\_regression\_points, 22

get\_regression\_points(), 24, 25

get\_regression\_summaries, 23

get\_regression\_summaries(), 23, 25

get\_regression\_table, 24

get\_regression\_table(), 23, 24

gg\_parallel\_slopes, 25

ggplot(), 16, 19

ggplot2::ggplot(), 26

glance(), 24

house\_prices, 27

ipf\_lifts, 28

aes(), 16, 19

alaska\_flights, 3

almonds\_bowl, 4

almonds\_sample\_100, 5

amazon\_books, 5

as.data.frame, 26



key glyphs, [17](#), [20](#)

layer position, [17](#), [19](#)  
layer(), [17](#), [19](#), [20](#)

MA\_schools, [30](#)  
ma\_traffic\_2020\_vs\_2019, [31](#)  
mario\_kart\_auction, [29](#)  
mass\_traffic\_2020, [30](#)  
moderndive, [32](#)  
moderndive-package (moderndive), [32](#)  
movies\_sample, [34](#)  
mythbusters\_yawn, [34](#)

nycflights13::airlines, [4](#)  
nycflights13::airports, [4](#)  
nycflights13::flights, [4](#), [12](#)  
nycflights13::planes, [4](#)  
nycflights13::weather, [4](#), [13](#)  
nycflights23::airlines, [13](#)  
nycflights23::airports, [13](#)  
nycflights23::flights, [12](#), [13](#)  
nycflights23::planes, [13](#)  
nycflights23::weather, [12](#), [13](#)

openintro::evals(), [14](#)  
openintro::gender\_discrimination(), [37](#)  
orig\_pennies\_sample, [35](#)  
orig\_pennies\_sample(), [36](#)

pennies, [35](#)  
pennies(), [35](#)  
pennies\_resamples, [36](#)  
pennies\_sample, [36](#)  
pennies\_sample(), [36](#)  
promotions, [37](#)  
promotions(), [38](#)  
promotions\_shuffled, [37](#)

saratoga\_houses, [38](#)  
stats::cor(), [21](#)

tactile\_prop\_red, [39](#)  
tidy(), [25](#)