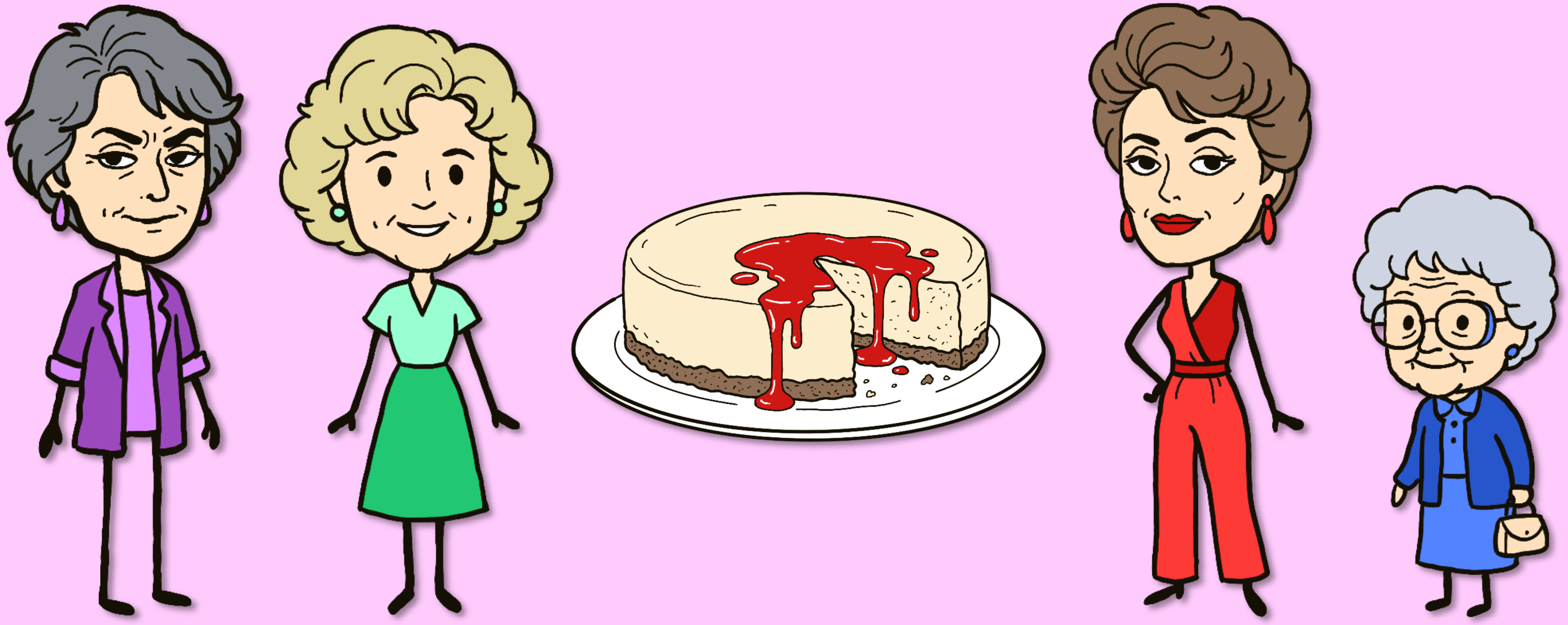
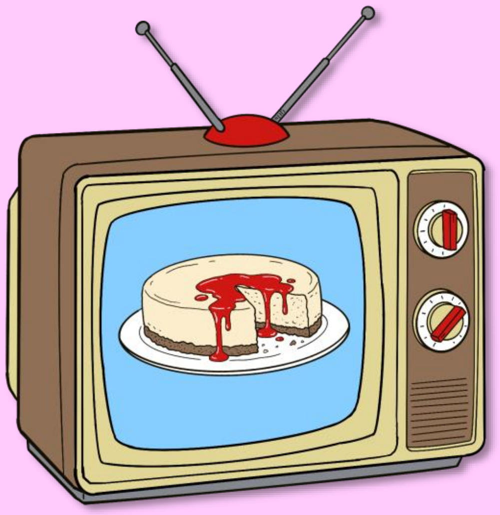
The Cheesecake ROI

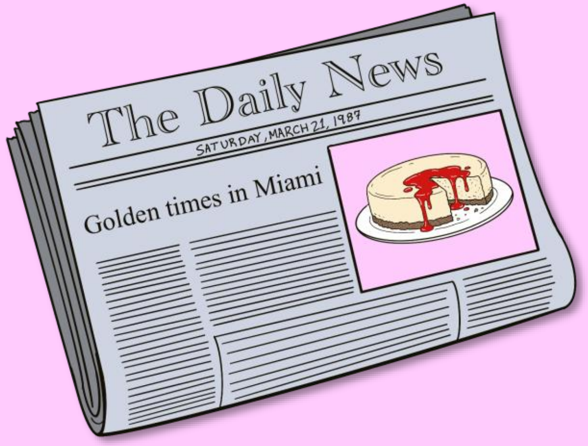
Tidymodels, Marketing Mix Models & The Golden Girls

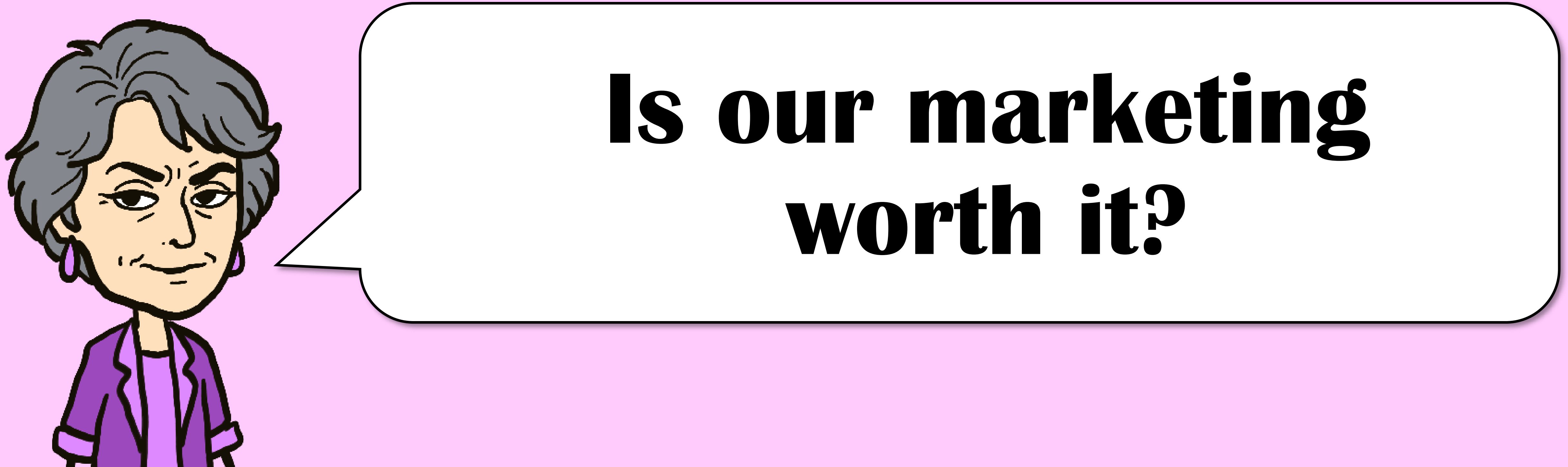
rainbowR conference 2026

Ryan Timpe

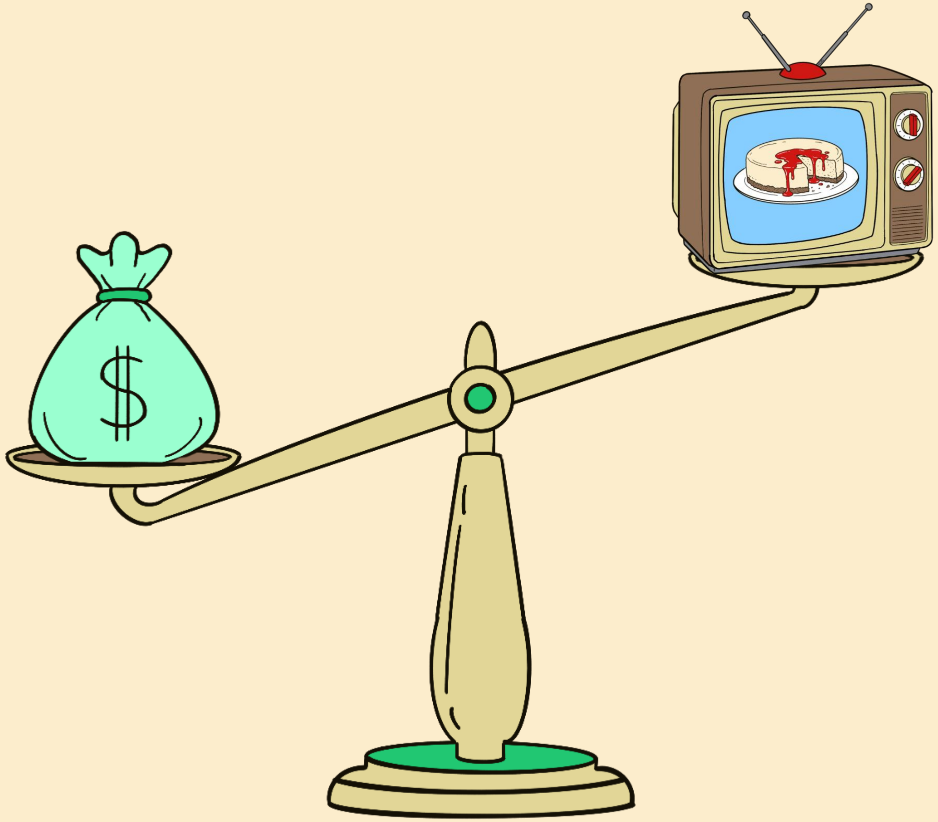





The Daily News
SATURDAY, MARCH 21, 1987
Golden times in Miami


Is our marketing worth it?

Return on Investment

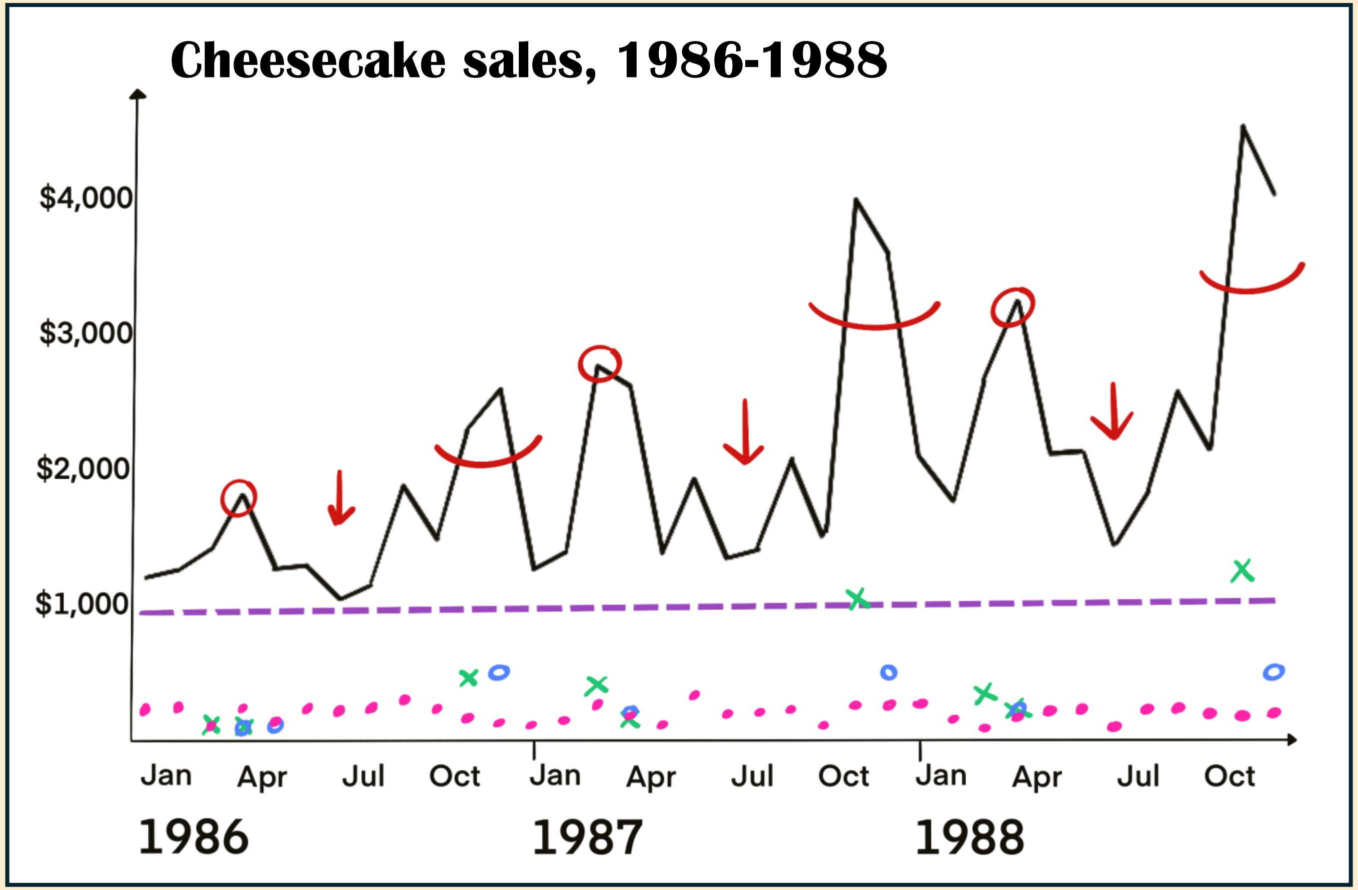


$$\frac{\text{Marketing-driven sales}}{\text{Spend}}$$

We know this.

We need to learn this.

Marketing Mix Modeling



- Baseline sales / intercept
- Holiday peaks
- Summer dip
- TV marketing
- Billboard marketing
- Newspaper marketing

Dataset

Year	Week	Sales	Season-ality	TV	Bill-board	News-paper
ID	ID	Y	Categorical	Reach	Reach	Reach
1986	1	$168		0	0	0
1986	2	$302		0	0	53
1986	3	$293		0	0	47
…	…	…	…	…	…	…
1986	47	$494		478	0	0
1986	48	$600	Thanksgiving	0	1000	7553
1986	49	$555		0	1176	12429
1986	50	$627	Christmas	0	1053	4775
1986	51	$626	Christmas	0	1111	2570

Base R – lm()

```
m1_lm <- lm(
  sales ~ .
  , data = cheesecake_data |>
           select(-time_year, -time_week))

summary(m1_lm)
```

```
Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)               280.46      12.32    22.8   <2e-16 ***
seasonalityChristmas       88.09      41.74     2.1     0.04 *
seasonalityEaster          37.16      53.18     0.7     0.49
seasonalitySummer         -72.94      15.81    -4.6    8e-06 ***
seasonalityThanksgiving   244.49      52.93     4.6    8e-06 ***
tv                          3.08       0.12    26.8   <2e-16 ***
billboard                   1.33       0.19     7.0    7e-11 ***
newspaper                   2.51       0.18    14.2   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 80 on 148 degrees of freedom
Multiple R-squared:  0.9,    Adjusted R-squared:  0.9
F-statistic: 2e+02 on 7 and 148 DF,  p-value: <2e-16
```

Tidymodels

What they say:

The tidymodels framework is a collection of packages for modeling and machine learning using tidyverse principles.
- tidymodels.org

What this means:

- Consistent syntax across different modeling methodologies and libraries.
- Build models using the |> pipe.

Tidymodels Structure

```r
library(tidymodels)

# Model Engine
model_eng <- linear_reg() # Defaults to "lm"

# Recipe
model_rec <- recipe(sales ~ .,
                    data = cheesecake_data) |>
  update_role(time_year, time_week, new_role = "ID")

# Join with a workflow
model_wf <- workflow() |>
  add_model(model_eng) |>
  add_recipe(model_rec)

# Fit
fit(model_wf, data = cheesecake_data) |> tidy()
```

Engine:

Libraries & type of model

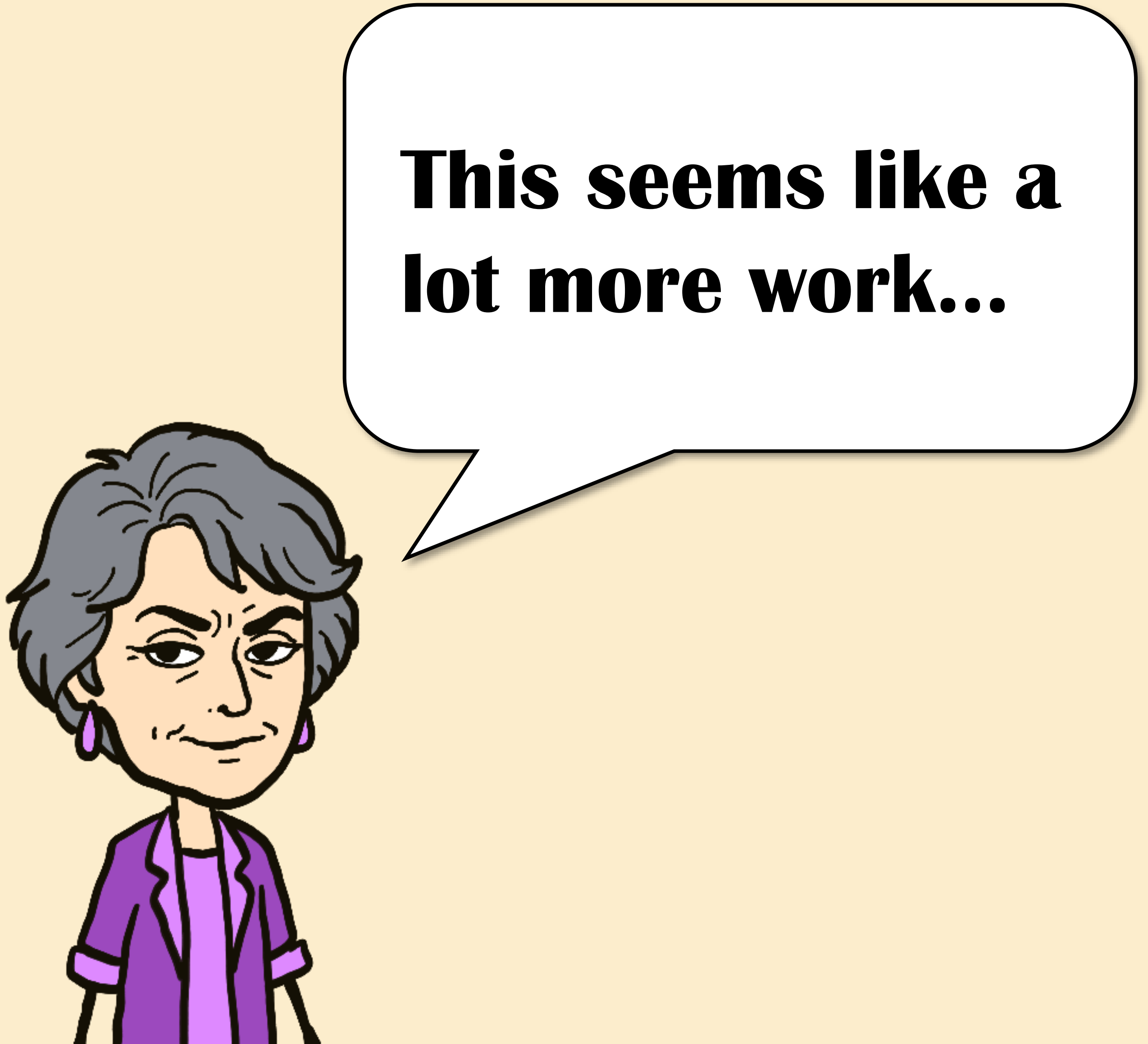
Recipe:

Formula, data, & transformations

Workflow:

Join, organize, & control process.

Tidymodels



Structure

Engine:

Libraries & type of model

Recipe:

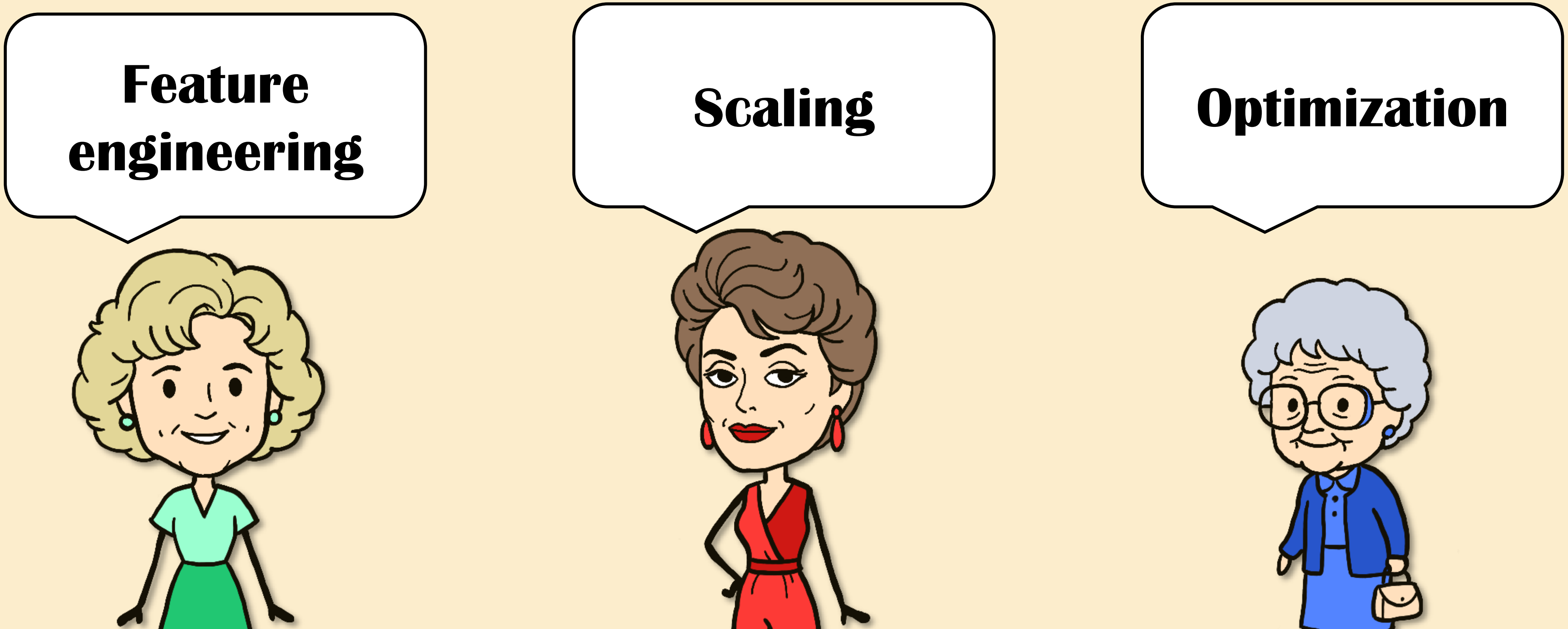
Formula, data, & transformations

Workflow:

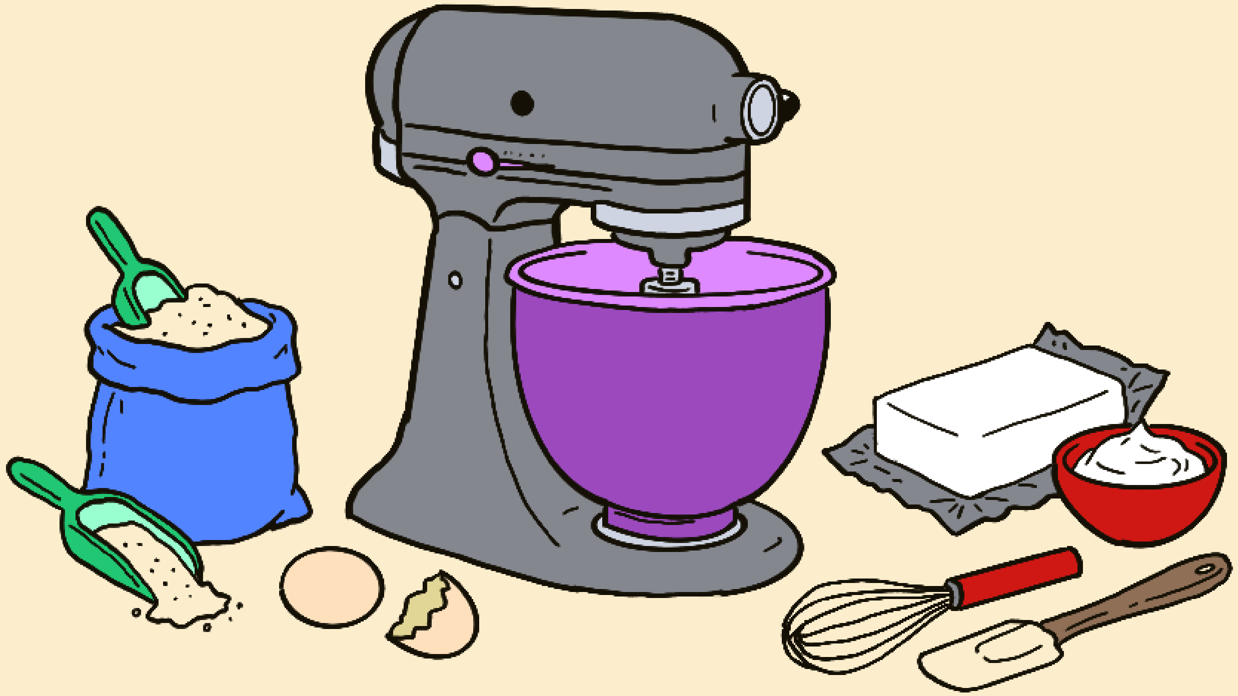
Join, organize, & control process.

Marketing mix models get complex

Tidymodels provides the structure to evolve simple models into powerful MMMs for deeper marketing insights.



Feature engineering



Feature engineering

- Recipes ensure that data processing and transformations are a part of the modeling process.
- Step_*() functions are key here.
- Metadata retained from training data to production.

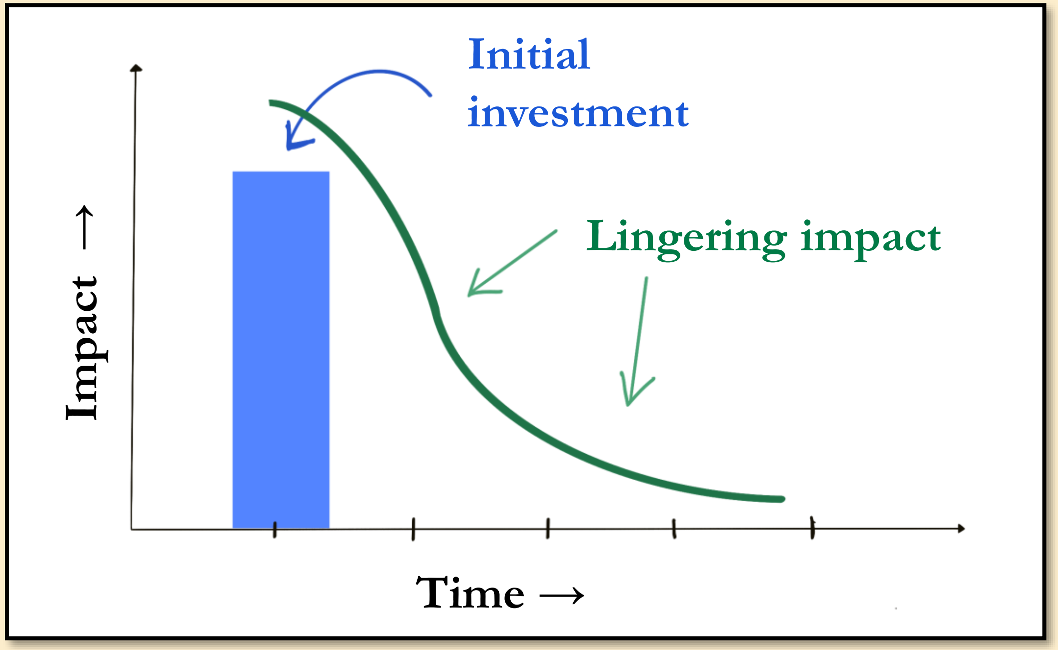
Learn more at recipes.tidymodels.org

```r
#Recipe with step_*() functions
model_rec <-
  recipe(sales ~ .,
         data = cheesecake_data) |>
  update_role(time_year, time_week,
              new_role = "ID") |>

  # Explicitly convert from categorical to booleans
  step_dummy(seasonality) |>

  # Min-max scale marketing
  step_range(tv, billboard, newspaper,
             min = 0, max = 1,
             clipping = FALSE)
```

Carryover impact

Marketing exposure at one point in time continues to have impact in future weeks.

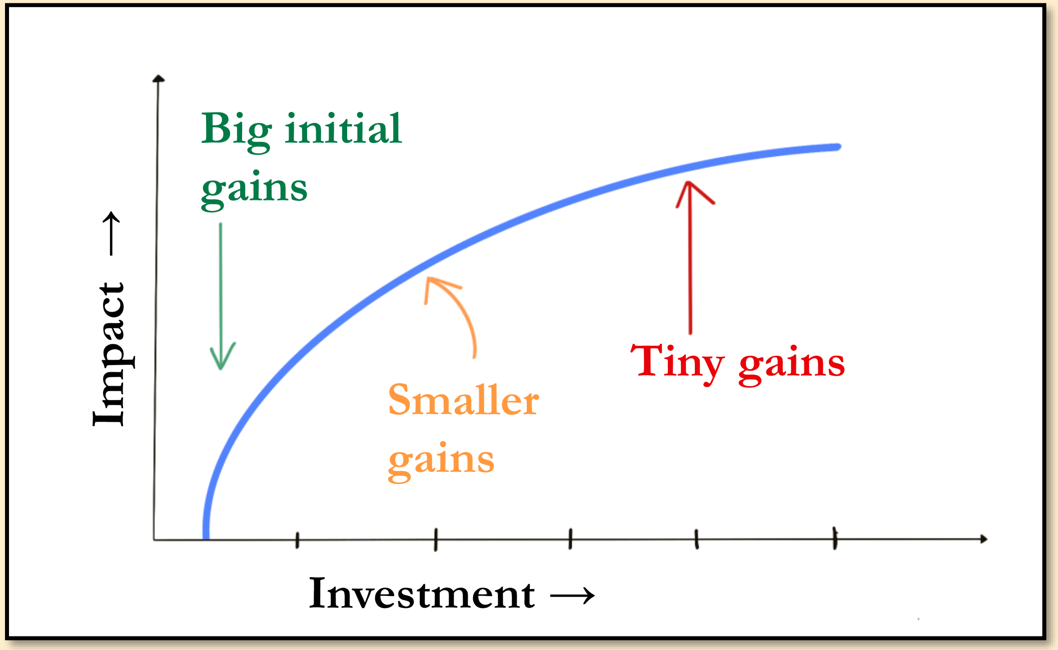
Observations are not independent.



Diminishing returns

High investment in marketing hits a saturation point and becomes less effective.

Relationships are not linear.

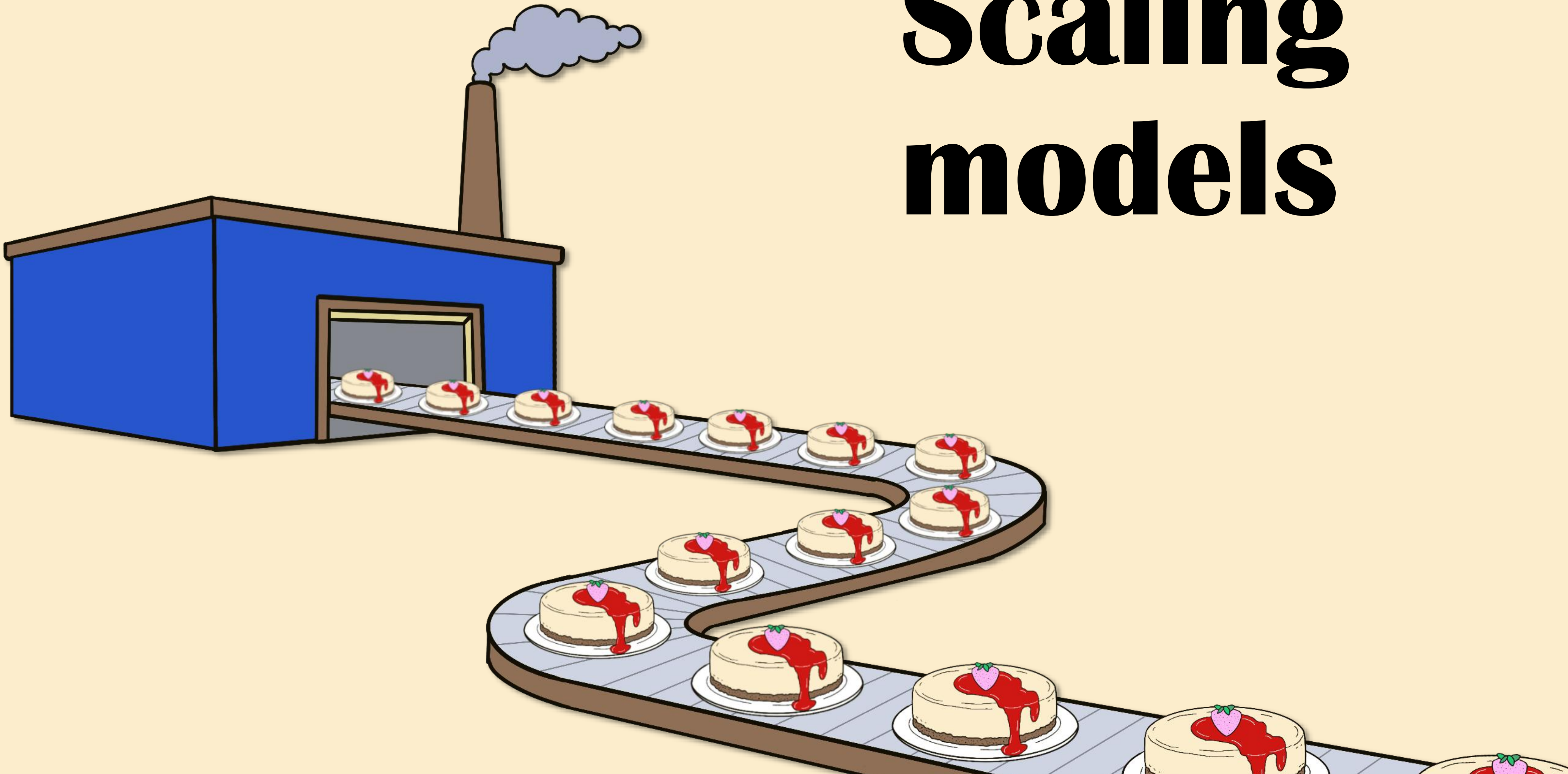


Carryover impact & Diminishing Returns

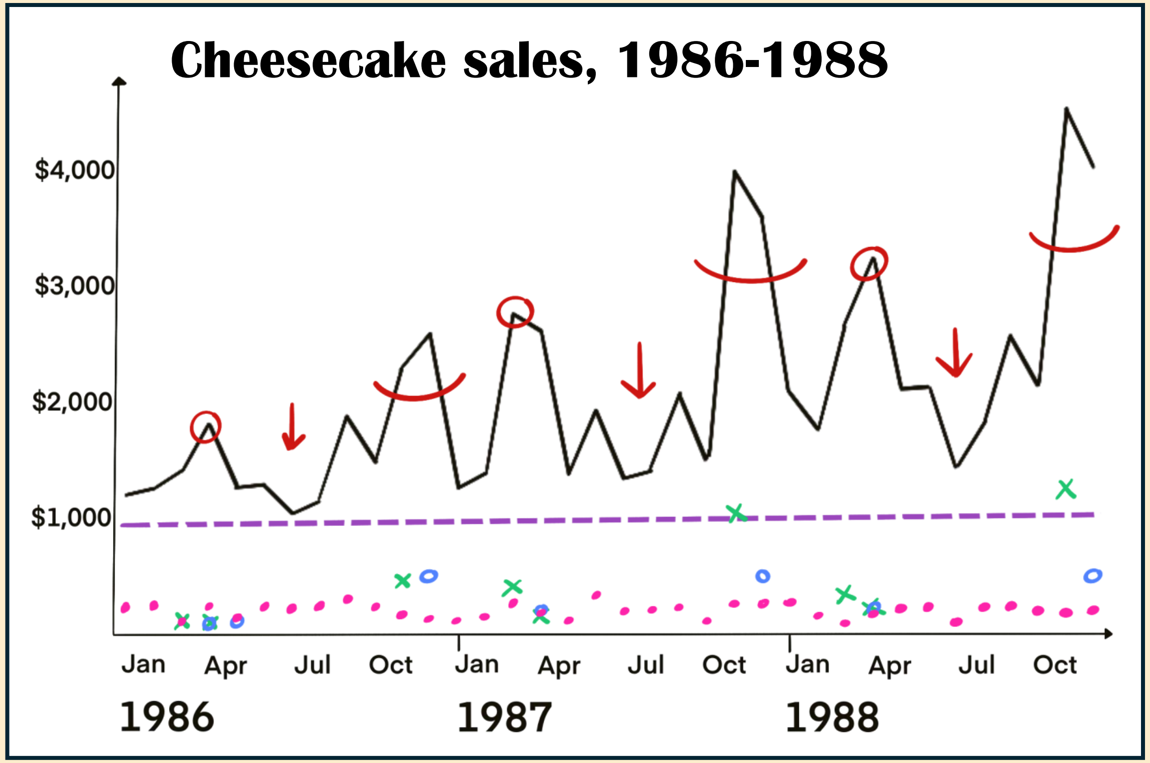
- Build custom step functions to capture these specific data transformations.
- Referencing existing code for a similar function or using LLMs is a great starting place.
- Order of operations matters.

```r
model_rec <- model_recipe |>

  # Carryover impact, specified for each input
  step_adstock(tv, billboard, carryover = 0.3) |>
  step_adstock(newspaper, carryover = 0.1) |>

  # Diminishing returns for all marketing
  step_power(tv, billboard, newspaper,
             power = 0.9) |>

  # Order of operations matter
  step_range(tv, billboard, newspaper,
             min = 0, max = 1, clipping = FALSE)
```

Scaling models



Scaling out



- Marketing inputs are correlated with each other.
- Marketing increases during holiday periods.
- As the business expands and model inputs grow, the default lm() might start failing.
- Factor analysis & PCA don't help in an MMM.

Model engine

Change the model engine while maintaining the same data, recipes, and syntax.

```r
# Model Engine
model_eng <-  linear_reg() |>
  set_engine("stan")

# Join with a workflow
model_wf <- workflow() |>
  add_model(model_eng) |>
  add_recipe(model_rec) #Same as before

# Fit
fit(model_wf, data = cheesecake_data) |> tidy()
```

Engine-specific parameters can be used, including priors for Bayesian models.

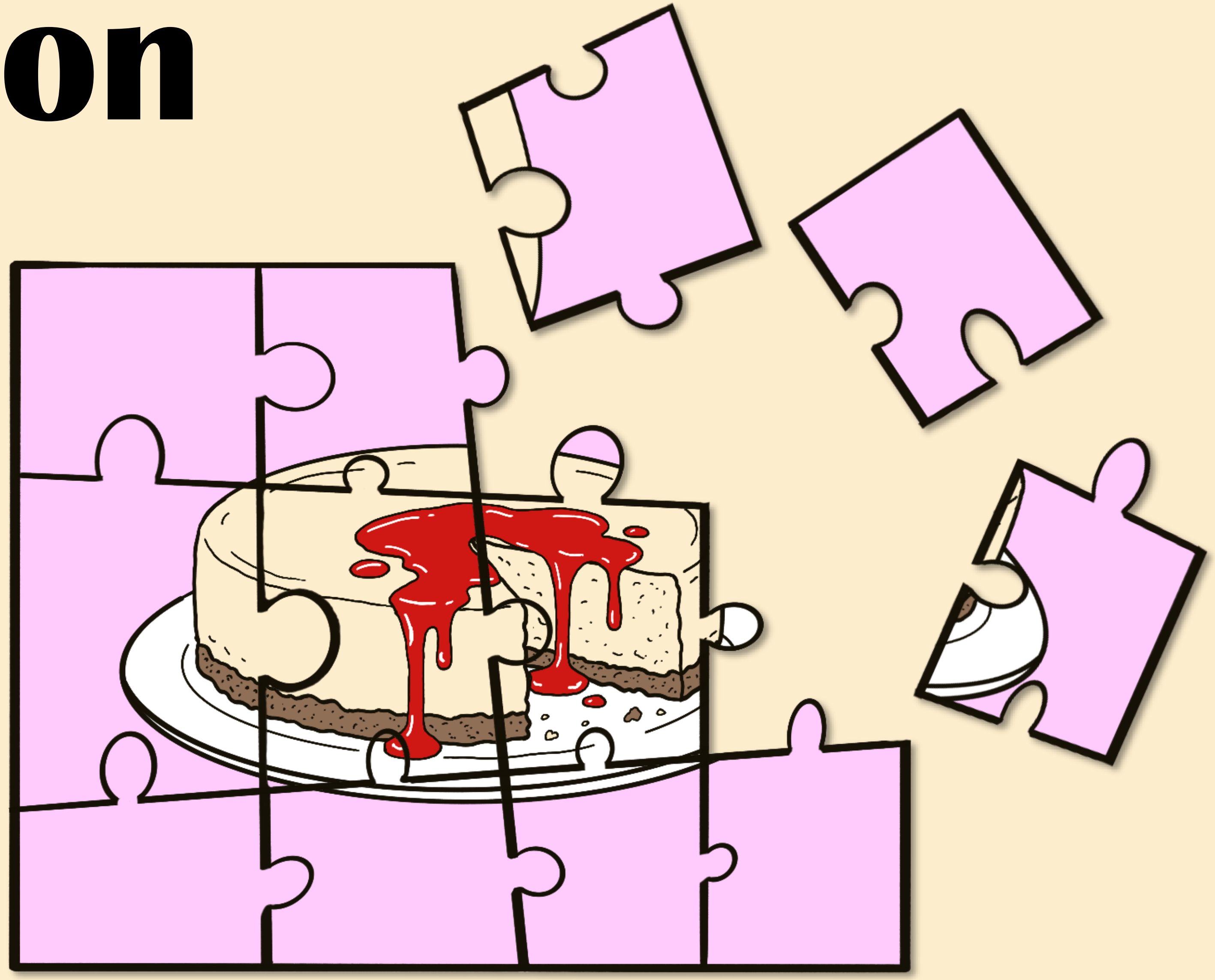
```r
model_priors <- rstanarm::normal(
  # Mean
  location = c(100, 100, -100, 100, #Seasonality
               100, 100, 100),  #Marketing
  # Std Dev
  scale = 50)

model_eng <- linear_reg() |>
  set_engine(engine = "stan",
             prior = model_priors)
```

Scaling up

- Programming with tidymodels.
- Wrap functions around workflows allow for flexible recipes.
- Use `any_of()` to reference inputs.
- Run many permutations of the same model. Use `purrr::map()` to organize runs.

```r
# Define function to run model with different marketing
run_mmm <- function(marketing_inputs){
  # Model engine

  # Recipe
  model_rec <- recipe(cheesecake_data) |>
    update_role(time_year, time_week, new_role = "ID") |>
    update_role(sales, new_role = "outcome") |>
    update_role(seasonality, new_role = "predictor") |>

    #Marketing inputs defined in function call
    update_role(any_of(marketing_inputs),
                new_role = "predictor") |>

    # Diminishing returns for all marketing
    step_power(any_of(marketing_inputs),
               power = 0.9)

  # Join with a workflow
  # Return the fit()
}

# Model with TV and Newspaper only
run_mmm(c("tv", "newspaper"))
```

Optimization



Optimizing inputs

- Dials package helps find the best parameters for feature engineering.
- Replace parameters in a recipe with `tune()`.
- Provide bounds for the inputs and acceptance criteria.
- Tidymodels includes tools for grid search or Bayesian tuning.

```
  step_power(tv, power = tune("tv")) |>
  step_s_curve(newspaper, saturation = tune("newspaper")) |>
```

```
# Define grid search range
marketing_tune_grid <- expand.grid(
  tv = seq(0.75, 1, by = 0.05),
  newspaper = seq(0.7, 0.9, by = 0.05))


# Grid search over workflow
model_res <- tune_grid(model_wflow,
                       # CV data with rolling_origin()
                       resamples = cheesecake_resamples,
                       grid = marketing_tune_grid)

# Top performing
show_best(model_res, metric = "rmse")

model_res |> collect_metrics()
```

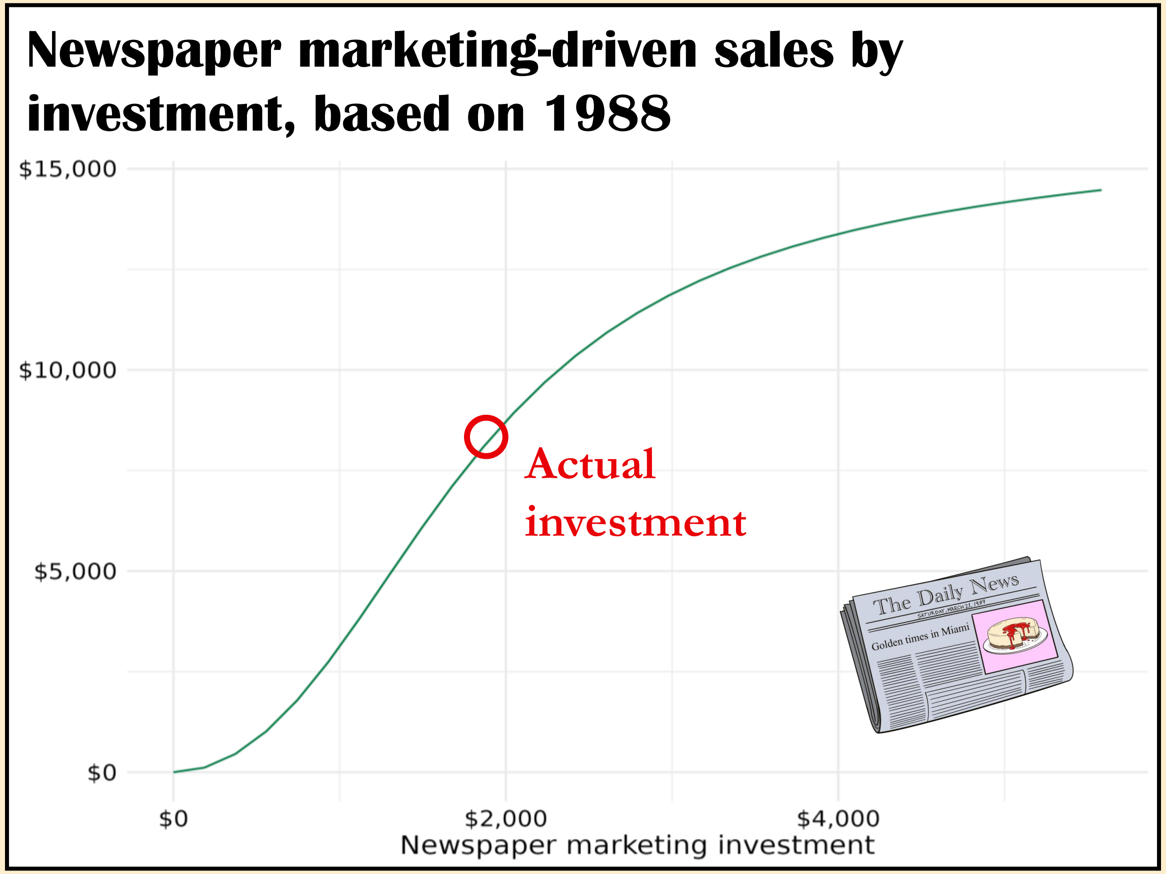
Optimizing marketing & response curves

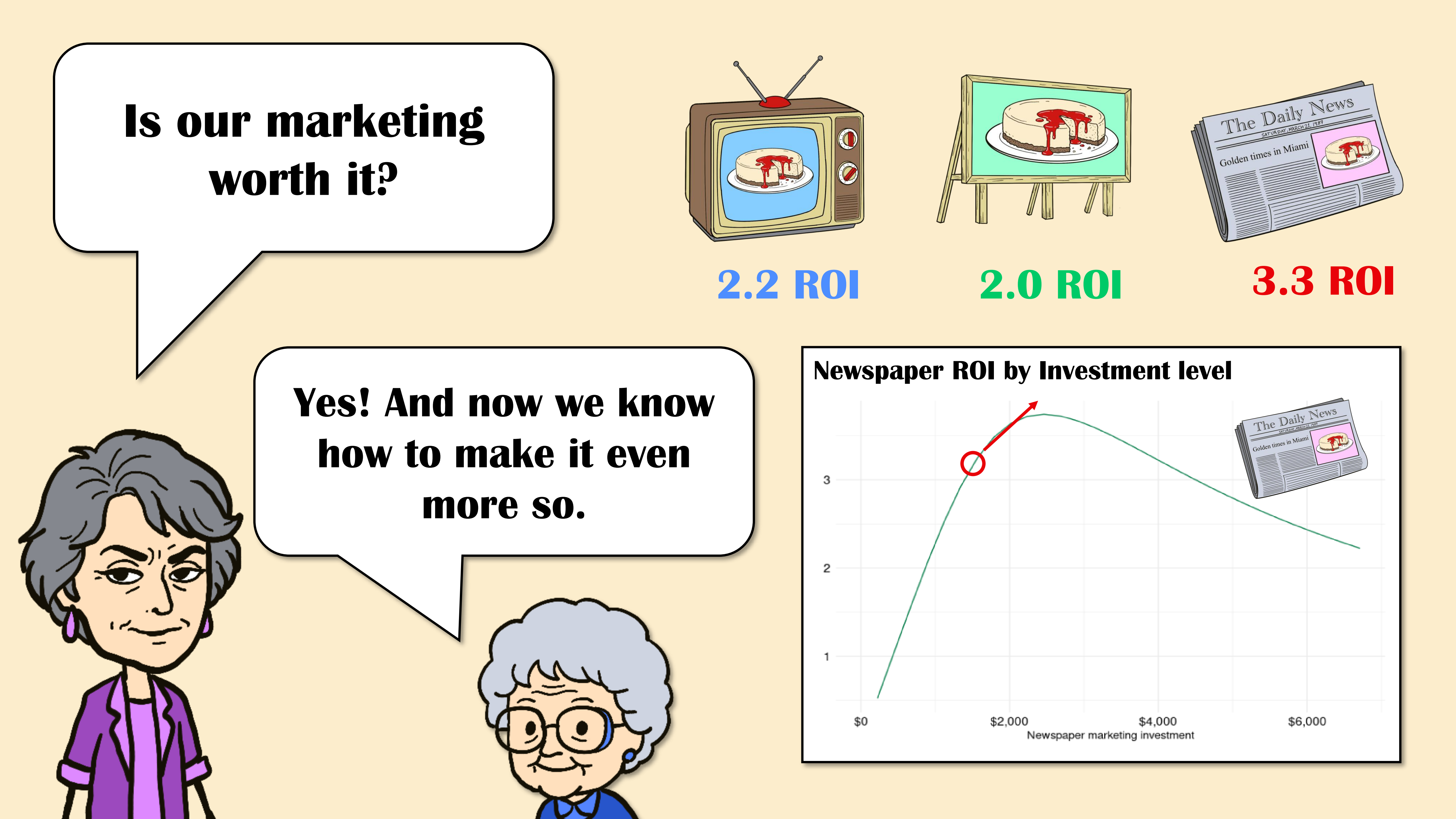
- Pass alternative datasets into trained models to build scenarios.
- Recipes retain properties from training, ensuring feature engineering is consistent in all scenarios.

```r
# Alternative scenarios change 1988 newspaper spend
# ... from 0% to 300%
# No feature engineering, just raw inputs

newspaper_scenarios <- seq(0, 3, by=0.1) |>
  purrr::map(\(xx){
  cheesecake_data |>
    filter(time_year == 1988) |>
    mutate(newspaper = newspaper * xx) |>
    mutate(scenario = xx)
})

# Pass scenarios and the trained model into predict()

newspaper_scenarios |>
  purrr::map( \(x){
    model_fit |>  # Trained model workflow
      predict(new_data = x) |>
      mutate(scenario = x$scenario[1])
  }) |>
  bind_rows()
```

Optimizing marketing & response curves

- Pass alternative datasets into trained models to build scenarios.
- Recipes retain properties from training, ensuring feature engineering is consistent in all scenarios.
- Use scenarios to build marketing response curves or find combinations of marketing that maximizes sales.





Is our marketing worth it?
The Daily News
Golden times in Miami
2.2 ROI
2.0 ROI
3.3 ROI
Yes! And now we know how to make it even more so.
Newspaper ROI by Investment level
3
2
1
$0
$2,000
$4,000
$6,000
Newspaper marketing investment

Thank you!

Learn more!

- Tidymodels.org
- Tidy Modeling with R
  by Max Kuhn & Julia Silge
- Ryantimpe.com
- Posit DS Community Discord