

# THE MELBOURNE DECONSTRUCTION

Capstone Project

February 27, 2018

## INTRODUCING THE PROBLEM

In my capstone project, I wanted to answer the following **research question**:

At my local coffee shop, are **sales of drinks** influenced by **weather**?

The shop's owner was excited to see if that is the case and provided me with daily sales reports in `*.csv` format—448 of them.<sup>1</sup>

These reports were generated by a cloud-based cash register system.

The system records sales of items like “latte,” “chai latte,” or “dirty chai”—names that the manager chose.

Third wave coffee aficionados aside,<sup>2</sup> most readers won't be able to explain **what makes these drinks different**. (And we're not even going in too deep yet.)

**Well, the data I received couldn't do that either.**

The client supplied me with daily sales reports counting unclear categories.

*How could I reinterpret the data to make it useful?*

I wanted to know if there's a relationship between what drinks the café sells and the weather conditions outside.

But what's the worth in knowing that, say, increased *pressure* and *temperature* are associated with *more sales of “lattes” or “mochas”*?

Knowing the answer wouldn't deliver much value to the client in terms of logistics or insight into consumer behavior.

Further, interpreting the results would be a nightmare. If we see a trend for “double-shot lattes,” but not regular “lattes,” is it the extra shot of espresso that matters? What if we don't see this same trend for “double-shot espressos,” which also have the extra shot but nothing else?

The problem is that **by default, a drink is just a name**.

It's an “americano,” a “latte,” a “cappuccino,” or a “macchiato.”

It has to be made meaningful, and the way to do it is to learn *what it's made of* and *how*.

To produce actionable insights, I'm not thinking in terms of *drinks*.

I'm thinking in terms of their *ingredients and traits*.

---

<sup>1</sup>To respect the conditions of my non-disclosure agreement with the owner, I will not name the business. This means that I will not be able to share or publish the raw files, as they contain identifying information.

<sup>2</sup>The client is part of **the third wave coffee movement** in the coffee industry: an independent-minded, artisanal reaction to the corporate **second wave** of the likes of Starbucks. The third wave coffee market treats coffee as a premium product defined by attention to detail in all stages of the process, from sourcing fair-trade beans to roasting to preparing quality espresso. By definition, these small businesses do not have access to big data that second-wave chains like Starbucks have. However, lately, the third wave industry has **attracted** large-scale investments. Putting aside the question of whether an overgrown third-wave coffee company can still be considered third wave, any third-wave coffee business will benefit from analyzing data available to it to thwart competition from both big chains and other small businesses.

Then, a **drink becomes a unique combination of ingredients**—espresso shots, portions of regular or specialty (almond, coconut, soy) milk, tea concentrates (chai or matcha), or smidges of chocolate. And don't forget **preparation techniques** (frothing to varying degrees) and **serving methods** (chilled or on ice).

Immediately, there's more precision to our thinking.

This approach fundamentally changes how sales should be recorded.

Let's take the example of recording a sale for a drink called “*dirty chai*.”

In my client's cash register system, it's recorded as a menu item of the same name.

On the menu, there's also an item called “*chai latte*.”

By just looking at the names, we wouldn't be able to tell that a “*dirty chai*” is simply a “*chai latte*” with a shot of espresso.

But if we were to forget *names* and think of *ingredients and traits* instead, a recorded sale in the abstract category “*dirty chai*” would become a sale of *an espresso shot, a portion of chai syrup, and a portion of frothed milk*, all taken separately.

Remember “**deconstructed coffee**” from Australia that took the Internet by storm a couple of years ago?

A Melbourne coffee shop became infamous for serving lattes like this:



In the world of “*deconstructed coffee*,” the only way to say “thanks a latte” is passive-aggressive.

It doesn't appear so at first, but **this “deconstructive” approach is very useful.**

While this is no way to serve a product to a client, this is a perfect illustration of what the product looks like to the café's manager.

**Logistically speaking, every *ingredient* should be stocked, and machines should be operational to make sure the drink has proper *traits*.**

For example, take ~~what was supposed to be a~~ the long macchiato above.

Normally, it consists of *an espresso shot*, made longer with *hot water*, and *milk—the ingredients*.

The milk is also *frothed*—that’s *a trait*.

The manager will stock coffee beans, which become espresso after passing through the grinder and the espresso machine.

They will also restock the milk and make sure the frothing nozzle is always operational.

Aesthetics aside, the Melbourne barista’s deconstructive approach was the approach of a responsible manager.

It took into account both *the logistics of the business* and *the highly customizable nature of the product*.

**I took the same approach to my dataset.**

I was *not* concerned with how many “*long macchiatos*” the café sold.

I wanted to know how many *espresso shots and milk portions* went into the drinks sold.

I’ll call this approach ***The Melbourne Deconstruction***.

This approach, where a drink is understood not as an entity representative of *one* category, but a unique product coded according to *several* defining features—ingredients and traits—would

- **help my data speak the language of my client’s logistics** and deliver value beyond what is offered by their cash register system, which already tracks sales of drinks, but not the ingredients,
- **streamline the interpretation of results at the modeling stage**, and
- **make it easier to discuss my data with those unfamiliar with third wave coffee culture**.

It’s good to know that something that was **made fun of on the Internet** can be used to our advantage in a data science project.

**It was the perfect target until it became the perfect weapon.**

## DATA WRANGLING

### PLANNING THE DATASET

To get **the raw sales data**, I asked my client to generate \*.csv files via the cloud-based cash register system they use.

The client exported 448 files.

Each of the files detailed all orders made at the coffee shop in one calendar day.

My goal was to pull the sales data from all 448 files into a single data frame.

Then, I would combine it with hourly weather data.

Originally, I **proposed** that I would use *daily* weather data, combined with *daily* sales.

However, after giving it some more thought, I decided to create a dataset of *hourly* sales combined with *hourly* weather data.

My reason for this was threefold:

- **Unlike the weather, the café does not operate 24/7.** It’s closed for at least ten hours every day, but the weather is still out there. The gaps in time would make the daily resolution weather data unrepresentative of the daily sales.

- **Hourly data is better suited for an immediate snapshot of the situation.** When I think of getting a coffee, I'm not thinking "what is the weather going to be like today, on average?" I'm thinking, "is it nice enough out to go grab a coffee, right now?" The hourly resolution is more representative of the consumer's decision making.
- **It would make for more than ten times as many observations.**

## WRANGLING CAFÉ DATA

Since I cannot share the files I obtained from the client, I made [a file illustrating the formatting of the raw data](#).<sup>3</sup>

Looking at **the raw data's formatting**, we see that

- the first lines always contain the name of the business and the calendar day for which the file details recorded orders,
- a section with reference information follows, explaining the fields that detail order information, and
- orders follow until the end of the file, separated by a series of dashes.

And **for every recorded order, there is**

- *General information:* order number and time
- *Price and payment information:* order sub-total before tax, calculated sales tax charges, taxable order total, order total for tax exempt orders, order total, payment type—cash or debit, device used to record the order
- *Information on items ordered:*
  - ID of menu item, name of the menu item, and price (e.g., "4, Large Latte, 2.83")
  - ID of menu item modifier, name of the modifier, and (not in all cases) price (e.g., "3, Extra shot, 0.48" or "1, Milk option: regular")

Since my analysis would concern drinks only, I would need to discard the data on food and services.

I would also discard the data on prices and payment.

To write the initial wrangling procedure, I used `readLines()` to extract lines from one file.<sup>4</sup>

I would feed one of the \*.csv files to my script, making a character vector with a sequence of order numbers and menu items ordered.

After examining printouts of this vector for different files, I realized **I was facing three big problems:**

1. **Most drinks could be modified in many ways. When looking at a record of the order, there was no way to tell what's a menu item, and what is an option to a menu item.** Most menu items at the café can be ordered with different options, or modifiers. For example, a large latte can be made with an extra shot of espresso, or with specialty milk to replace regular milk. There could be any number of options recorded. Other than knowing that the first item in an order would not be a modifier, a script could not gauge by looking at the data which level of hierarchy each object is. The only unique thing about modifiers is that some of them don't have prices, but most do. When I extracted all the ID-name pairs from several files, I concluded that there was also no regularity to the identification numbers. They appear to have been in the order in which the café's manager recorded them into the system.

---

<sup>3</sup>The raw files I worked with were in French. The names of items were changed. "\$\$\$" replaces price information. The file is abridged to show what the beginning and end of a raw file would look like.

<sup>4</sup>A later version of this procedure can be found in a function [here](#).

2. **Identification numbers were double-booked, referring to both menu items and options.** For example, 4 would refer to “large latte” and also to “soy milk.” This means that the same number can refer to two objects, each on a different level of the item-modifier hierarchy.
3. **At least several number-descriptor combinations—typos and promos—referred to the same menu item.** Some menu items in the data are typos—a result of human error when maintaining the menu in the cash register system. For example, “CappucciNNo” is a typo: it should be recorded as “CappucciNo.” Still, because of the typo, it appears like a separate item called “CappucciNNo” was sold for a few days. Also, promotional versions of regular items were recorded as separate items. For instance, “ESPRESSO PROMO!!” isn’t a separate item and would need to be recorded as the rest of “espressos.”

My solution to the problem of discerning between menu items and modifiers, as well as typos and promos, was to create a *reference table* where every ID-descriptor pair is coded according to its type and position in the menu-modifier hierarchy.

I would then expand the reference table into a *coding book*.

The coding book would be the key to performing *the Melbourne Deconstruction*: recording each drink sale as a unique combination of ingredients and traits.

**I wrangled the client-supplied data in four steps**, each implemented in a separate R script:

1. **Learn what’s on the menu and put it into a reference table.** I first extracted all unique pairs of identification numbers and descriptors from all 448 files. Then, I manually coded each ID-descriptor pair according to its type (drink, food, or service), position in the menu-modifier hierarchy (noting whether it’s a standalone menu item, like a latte, or an option to a menu item, like soy milk), and correct spelling (if it was a typo or a promotional item). I would use this reference table later to differentiate among elements within a vector extracted from a raw \*.csv file.
2. **Extract data on all orders completed at the café into a data frame.** The data frame of orders includes ID number for the order, day, HH:MM, and the contents of the order stored as a string.
3. **Using the reference table, discard unnecessary items (food and services)** from the data frame of orders. After this was done, I expanded my reference table into a coding book. The coding book contains codes for every feature that I could potentially use in my statistical analysis. Most variables code TRUE or FALSE on the presence of a given ingredient or trait in a drink. For instance, a latte is coded TRUE on espresso, milk, and froth.
4. **Build a table of hourly counts of sales by drink feature.**

**The final script was particularly complex and included several stages:**

1. I **parsed** my *data frame of orders* made at the café (one observation is an order that may contain several drinks), making it into a *data frame of drinks* (one observation is one drink: the base menu item and modifiers).
2. I **replaced** typos and promos with the menu items’ original names.
3. Then, I **joined** the data frame of drinks with the coding book, adding the variables for coding the drinks on traits and ingredients.
4. Having the data (in string form) on what modifiers were used and the default codes of base drinks, I **overwrote** the default codes with relevant modifiers’ codes. For instance, having your latte “decaf” would cancel out the “espresso,” or having it “iced” would cancel out “frothed.”
5. Finally, I **cleaned** the resulting data frame, **counted** tallies for hourly sales of each ingredient and trait, **wrangled** the weather data for the client’s location, and **joined** it with the dataset of drink sales.

## WRANGLING WEATHER DATA

To obtain the weather data, I initially tried using the `weatherData` package, which pulls data from `wunderground.com`. However, the website changed the URL directory structure that the package relied on. When I was trying to pull the data, `wunderground.com` responded with “bad request,” and the issue `wasn't fixed` yet.

I considered other R packages, but most of them seemed to require getting API keys from the weather data providers they connected to.

This seemed unnecessarily time-consuming, compared to the option I `proposed` initially—using government-supplied data from Climate Canada.

The agency had `*.csv` files available for download with no special arrangements needed.

For the weather station in my client's location, reports were available at `daily` and `hourly` resolution.

For my purposes, downloading government data proved to be faster and more straightforward than using packages.

The wrangling procedures I performed on the weather data were less `nightmare-inducing` exciting than what I had to do for the sales data, but they can be found in `my last wrangling script`.

## DIVING INTO THE DATASET

### OVERVIEW

The dataset can be downloaded `here`.

Every observation in the dataset is *one hour of the cafe's operations*.

For example, an observation can be partially spelled out like this:

```
On September 1st, 2016, from 10:00 to 10:59AM,
the cafe sold 6 espresso-based drinks,
6 drinks containing milk, 1 drink containing chocolate,
0 drinks that were chilled or iced, and 7 frothed drinks.
During that time, the sky was clear, there were no fog, rain, or snow,
and the temperature was at 20.5 degrees C.
```

The next two sections describe the dataset's dependent and independent variables.

### DEPENDENT VARIABLES

All dependent variables other than *DrinksSold* and *Size.Mean* describe the number of drinks adhering to the noted feature that were sold during the hour of observation.

The following table provides context on the coding process, noting examples of drinks that were coded as TRUE on the relevant features (ingredients and traits).

Variable	Description
<i>DrinksSold</i>	Total number of drinks sold.
<i>Size.Mean</i>	Mean volume of drinks sold, in milliliters.
<i>Content.Water</i>	Water understood as a separate ingredient added to the drink (e.g., americanos are made by pouring an espresso shot in a cup of hot water). This excludes water going through an espresso machine, filter pot, chemex or other pour over devices.



Variable	Description
<i>Content.Tea</i>	Tea leaves, chai, and matcha, as well as kombucha (fermented tea).
<i>Content.RegularMilk</i>	Regular milk is the default option for making espresso-based beverages (e.g., lattes, cappuccinos). In a hot drink, the milk is frothed. In a cold drink, it's used as is.
<i>Content.SpecialtyMilk</i>	For an additional price, clients have the option to substitute regular milk for specialty milks (coconut, almond, or soy). Appeals particularly to the lactose intolerant. However, many others prefer the taste of a certain specialty milk to the regular milk.
<i>Content.Chocolate</i>	Found in hot chocolates (which have quite a large portion of chocolate) and mochas (lattes with a smidge of chocolate on the bottom to offset the bitterness of espresso).
<i>Content.Seasonal</i>	Seasonal ingredients added to standard drinks (e.g., latte), like pumpkin spice (fall) or mint (winter).
<i>Content.Juice</i>	Bottled juices.
<i>Trait.HighInSugar</i>	Drinks high in sugar, compared to other drinks offered at the shop. Primarily, drinks with chai syrup.
<i>Trait.HighInCaffeine</i>	Drinks with extra espresso shots or filtered/drip/slow bar coffee. For a discussion on the concentration of caffeine in espresso vs. drip coffee, see <a href="#">this post</a> .
<i>Trait.Cold</i>	Drinks served on ice, as well as cold brew coffee or any bottled or canned drinks served from the fridge.
<i>Trait.Froth</i>	Hot drinks with milk, prepared using a frothing nozzle on an espresso machine.

## INDEPENDENT VARIABLES

The dataset contains the following weather variables:

- **Temperature**, degrees Celsius
- **Dew Point temperature**, degrees Celsius
- **Relative humidity**, percent
- **Wind speed**, km/h
- **Pressure**, kilopascals
- Binaries for **atmospheric phenomena and sky conditions**—rain, snow, fog, and clear sky

## IMPLICATIONS

The dataset can have implications to several areas of the client's day-to-day operations:

- **Stock management.** Having insight into the effects of weather variables on product demand, the client can adjust stock management. For example, they could order less milk if they expect less demand for milk-based beverages given tomorrow's weather forecast.
- **Product development.** Knowing how weather conditions influence beverage preferences, the client may decide to adjust the current product selection or create new products. To illustrate, knowing that there is a better chance of selling a chocolaty beverage when it rains may lead to developing a special drink containing chocolate only sold on rainy days.
- **Marketing strategies.** Aware of just which products customers want to buy in certain weather conditions, the client may adjust what—and how—they market. Weather-based marketing won't just

involve deciding what witty message will work well on that sidewalk sign outside the door. For instance, the client may want to collect their customers' email addresses to produce weather-triggered marketing campaigns or send out a timely call to action on the business' social media accounts.

Importantly, the dataset can only be used to answer research questions about my client's business, and not the third wave café market or the coffee industry in general.

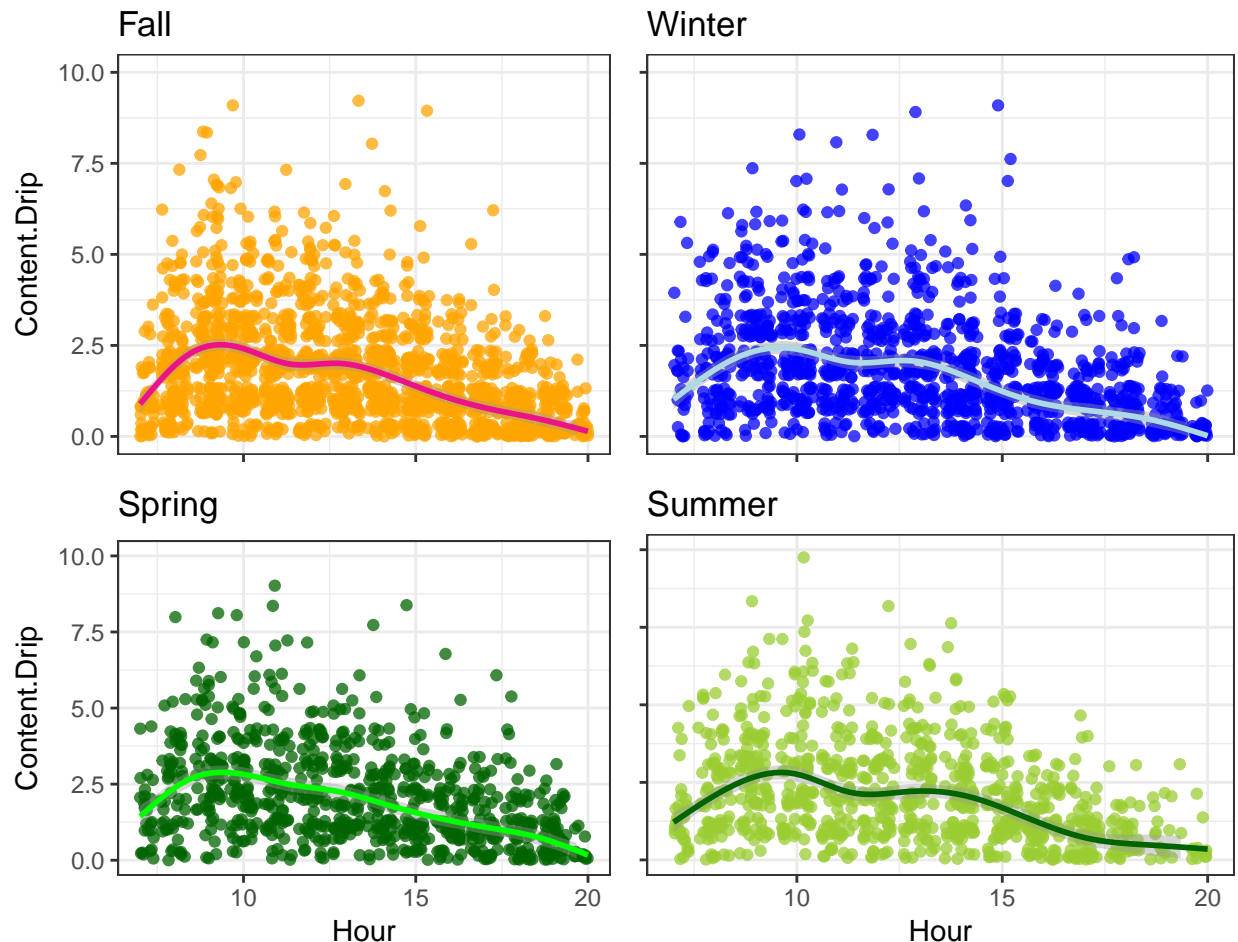
## PRELIMINARY EXPLORATION

While my inquiry will be focused on the relationship between weather and sales counts, the dataset can be useful to my client in other ways.

My dataset is cleaner and more nuanced than what the client has access to via their cash register system.

It may let us spot trends that the client may not have been aware of.

For instance, this plot shows the number of filtered coffee drinks sold each hour, by season:



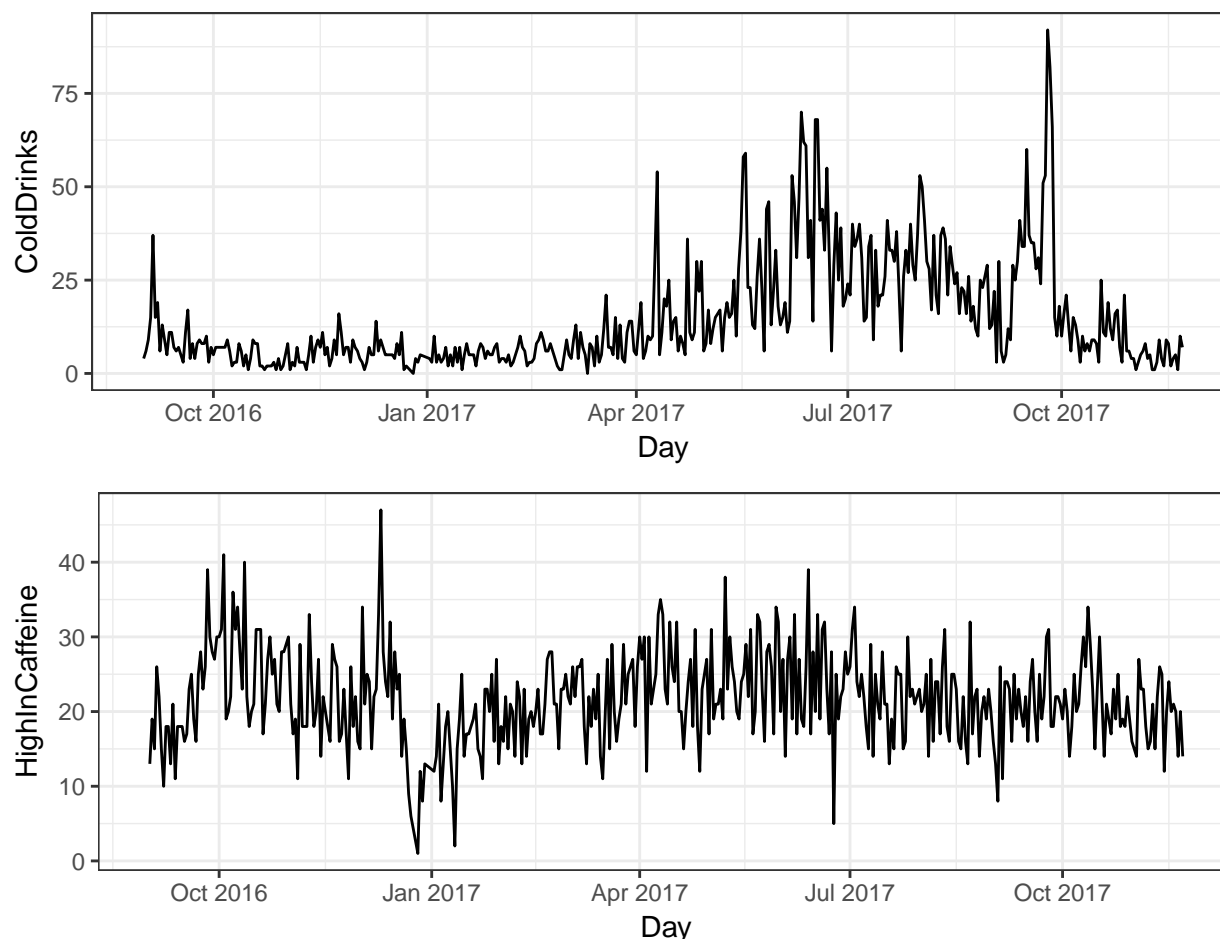
Judging by this, it's probably not a good idea to start brewing a pot of filtered coffee after 5PM, unless there are orders coming in—particularly in the summertime.

Finally, if needed, the dataset can be rescaled to the daily resolution, providing a further dimension of insight for the client.

```
daily <- dataset %>% group_by(Day) %>%  
  summarise(sum(Trait.Cold), sum(Trait.HighInCaffeine), mean(Temperature))
```



For example, we could plot daily **time series** for drinks with specific features, like *cold* or *high in caffeine*:



These are just some possibilities of what can be done with the dataset besides analyzing the impact of weather variables!

## STATISTICAL ANALYSIS

### HYPOTHESIS

There is a wealth of information to examine in the dataset: there are 15 dependent variables to choose from. However, I am also a novice in statistical analysis.

Given this, I will limit myself to analyzing just one variable.

I chose to examine *sales of cold drinks*<sup>5</sup> because:

- Commonsensically, there should be at least one sure-fire predictor for this variable—high temperature.
- As a consequence, I expect the results to be more easily interpretable: if higher temperature cannot predict sales of cold drinks, it will mean that I need to troubleshoot the model.

---

<sup>5</sup>As noted in the table detailing dependent variables, cold drinks are those served on ice, as well as cold brew coffee or any bottled or canned drinks kept in a refrigerator.

My **hypothesis** is this:

Higher numbers of cold drinks sold coincide with higher temperatures, better visibility (manifested by clear skies and lack of fog), and lack of precipitation.

## A NOTE ON MODELING

Although my dataset features *time-series count data*, I will be using a linear model to assess the effects of weather variables.

There are two reasons for this:

- **The appropriate statistical analysis tools—time series analysis, as well as Poisson and Negative Binomial Regression—are outside the scope of this workshop.** Given that I lack prior academic training necessary to apply the more complex models, I will have to ignore the time-series and count nature of the data and use linear regression.
- More importantly, **I am interested in the *directionality* of the relationships between the variables, rather than exact predictions.** Further, I want to know which *among* the weather variables have stronger effects.

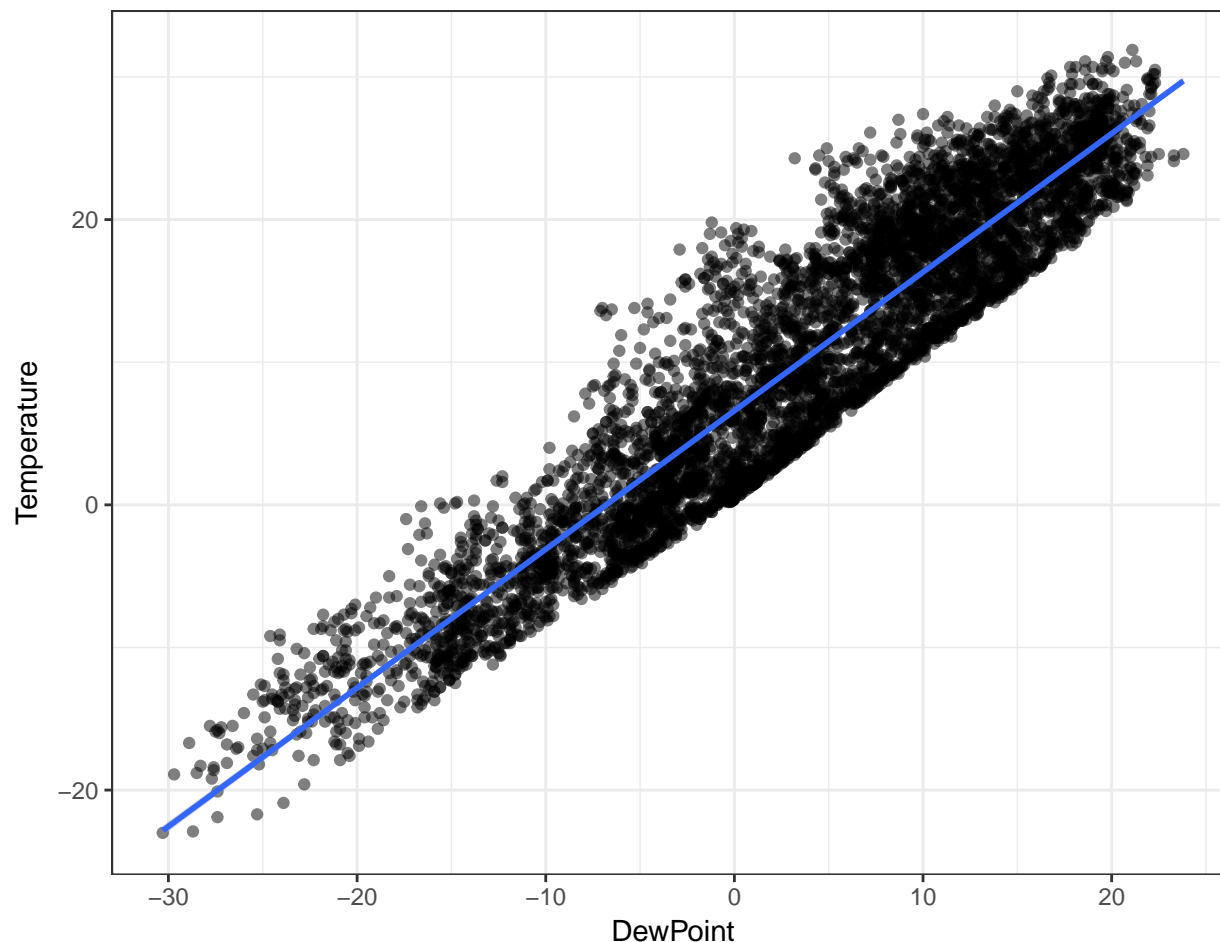
Given this, *I will treat each observation in my dataset as an independent snapshot, rather than a part of a time series.*

## CORRELATING INDEPENDENT VARIABLES

First, I will choose the maximum number of independent variables to use in the model.

It will be helpful to see whether any of the weather variables correlate among each other, so that I know what variables not to include in modeling to avoid multicollinearity.

The output of a correlation matrix shows that none of the variables' correlation coefficients are over 0.5, except for one pair: *temperature* and *Dew Point temperature* (0.93):



This is a very strong correlation, and Pearson's test confirms that it is significant:

```
##
##  Pearson's product-moment correlation
##
## data:  dataset$Temperature and dataset$DewPoint
## t = 193.57, df = 5665, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9285201 0.9353630
## sample estimates:
##          cor
## 0.9320246
```

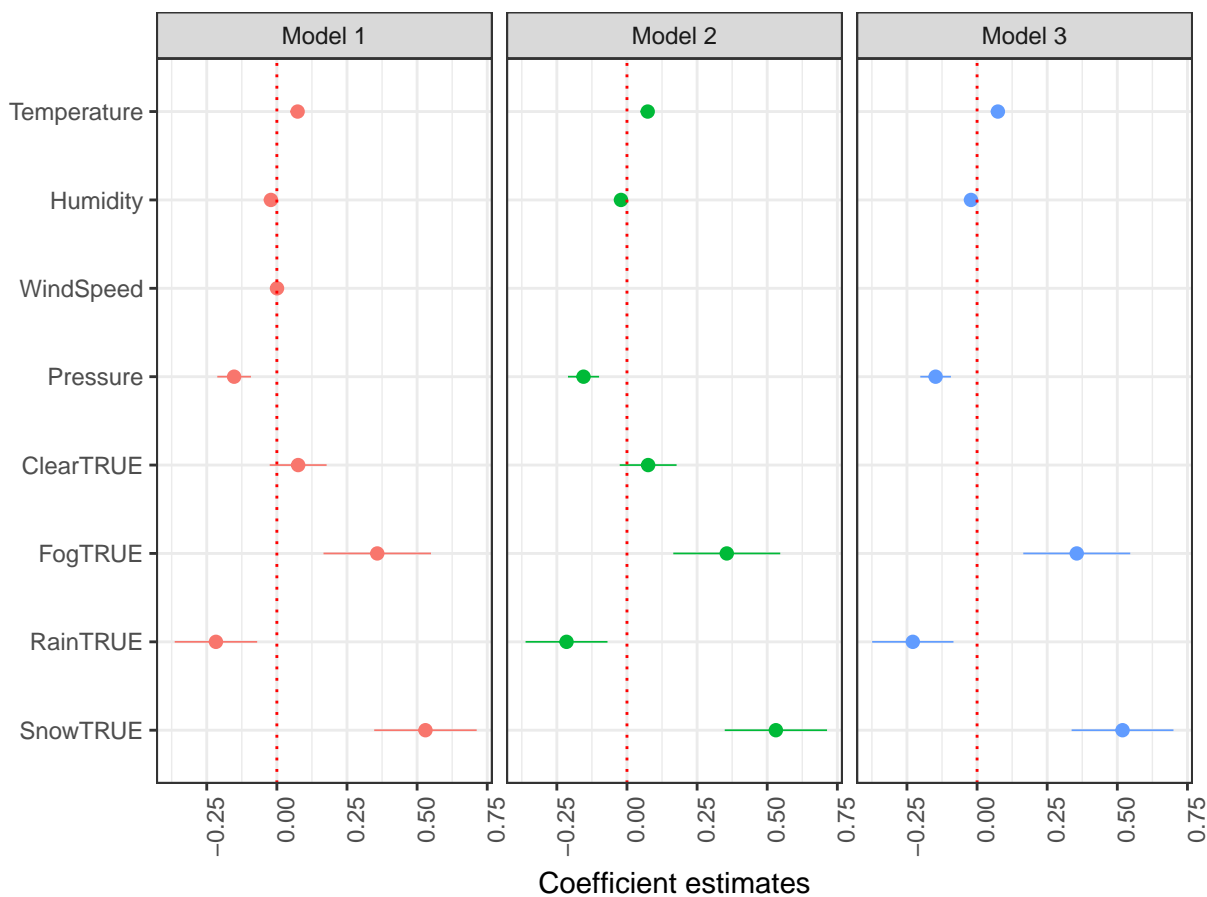
Given this, I will refrain from using *Dew Point temperature* as a predictor in my modeling.

## MULTIPLE LINEAR REGRESSION

I will start building a model using all independent variables besides *Dew Point temperature*.

Over two iterations of removing insignificant predictors (*wind speed* and *clear sky*), I arrive at three models:

## Predicting sales of cold drinks



A significant regression equation was found ( $F(6, 5660)=308.5$ ,  $p < .000$ ), with an  $R^2$  of .25.

Over each iteration, removing the insignificant predictors did not impact the remaining predictors, and  $R^2$  decreased only by 0.0002 between model 1 and model 3.

## CROSS-VALIDATION

I will need to compare the three models using cross-validation.

Running a leave-one-out cross-validation procedure on the three models shows that error estimates get smaller for each successive model, confirming that the final model is the best among the three:

```
L001$error
```

```
## [1] 2.60314
```

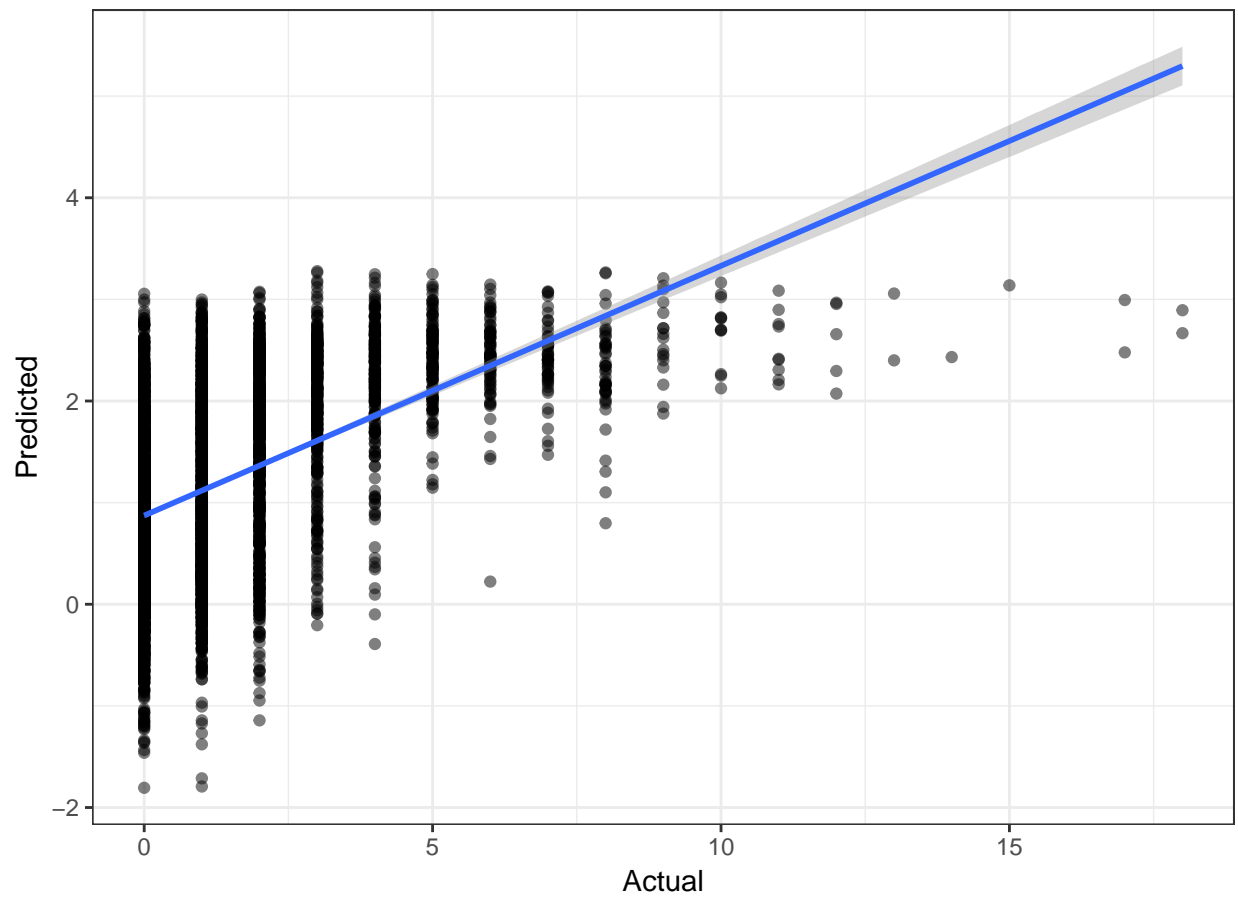
```
L002$error
```

```
## [1] 2.602215
```

```
L003$error
```

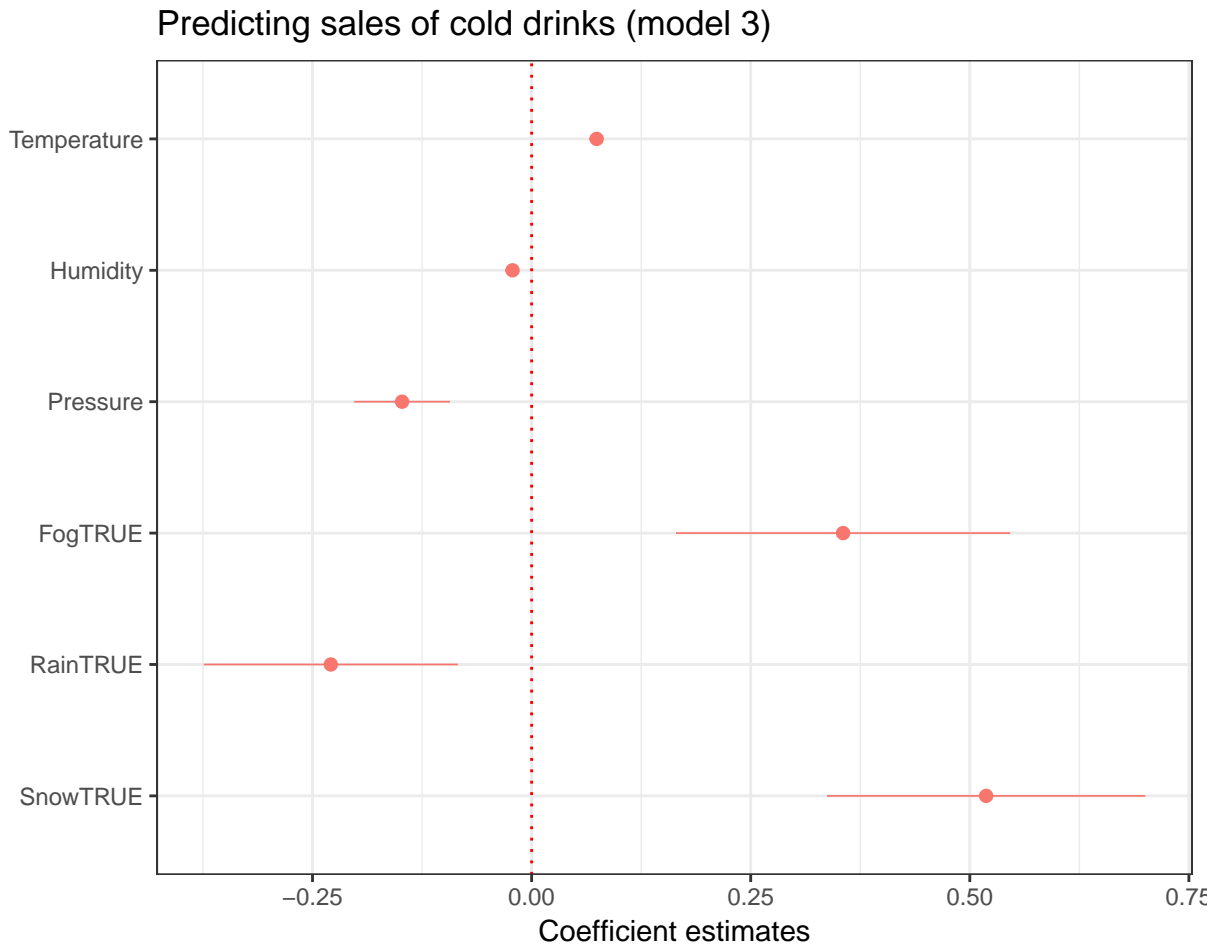
```
## [1] 2.602058
```

Actual vs. Predicted (model 3)



## TROUBLESHOOTING

I will now more closely examine the model for predicting cold drinks:



There's room for improvement in my model, because:

- The impact of the binary variables, as measured by coefficient estimates, seems unreasonably strong. This is particularly the case for *snow*, the coefficient estimate for which is far greater than for *temperature* or any other variable in the model.
- The three binary variables (*Fog*, *Rain*, and *Snow*) have suspiciously large confidence intervals, compared to the continuous variables.

I suppose that this may be due to **two possibilities**:

- the binary variables have *outlier observations* that influence their effects, or
- a *lack of precision in measurement* of the binary variables (*snow*, *rain*, and *fog*) amplifies their effects compared to the continuous variables (*temperature*, *humidity*, and *pressure*).

I explore these possibilities in the next section.

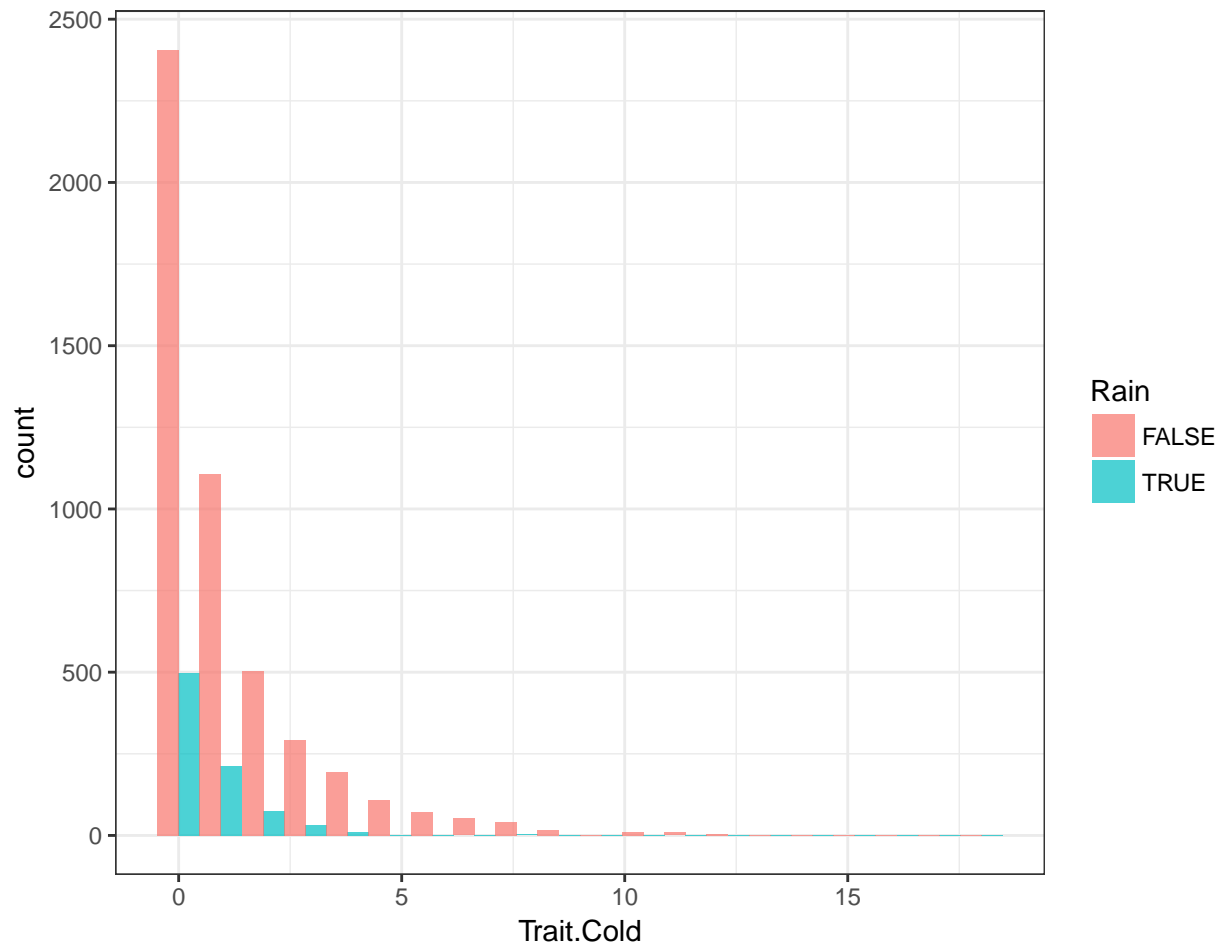
## OUTLIERS

### RAIN

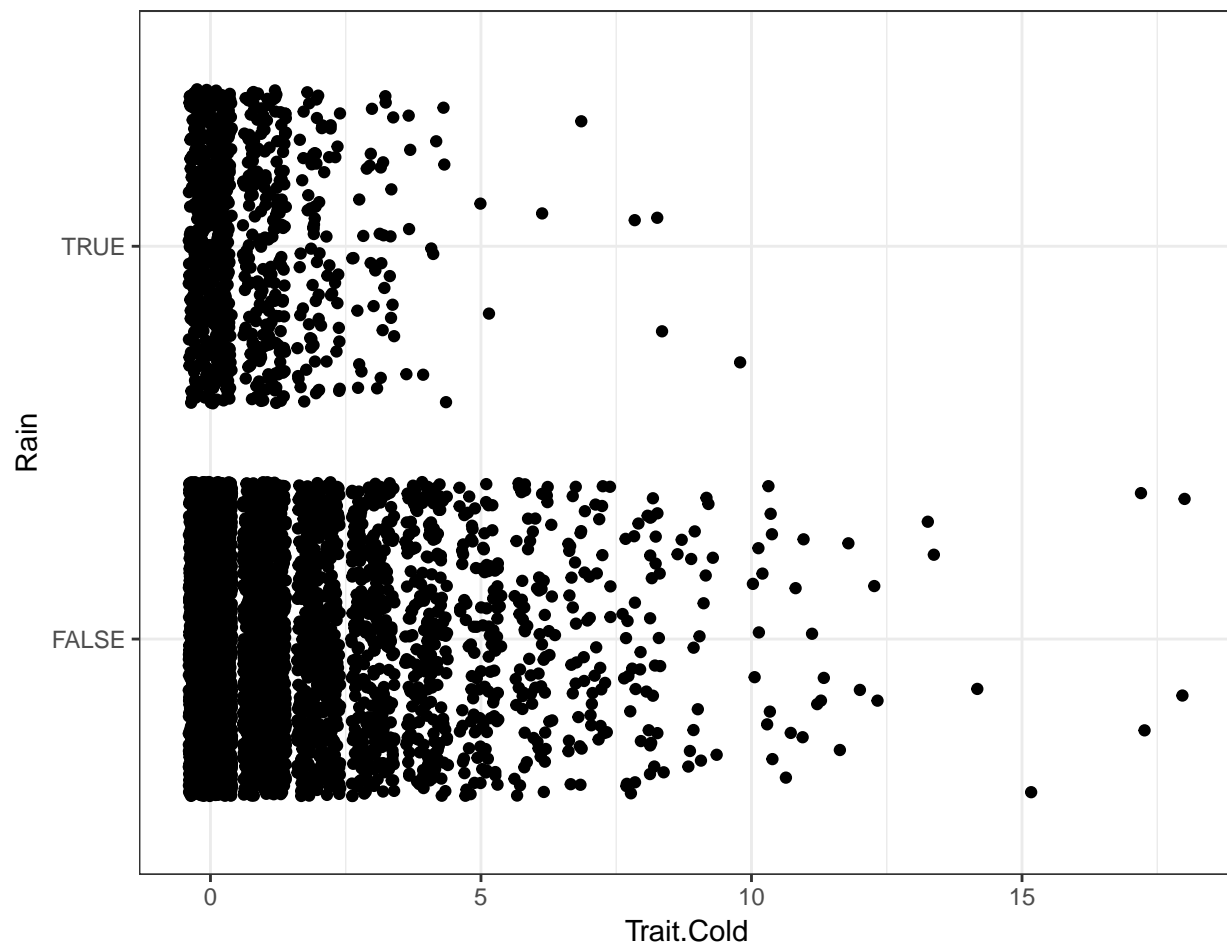
In the model, *Rain* is a negative predictor with a large confidence interval.

This seems reasonable, because during most of the rainy hours, no or very few cold drinks were purchased:

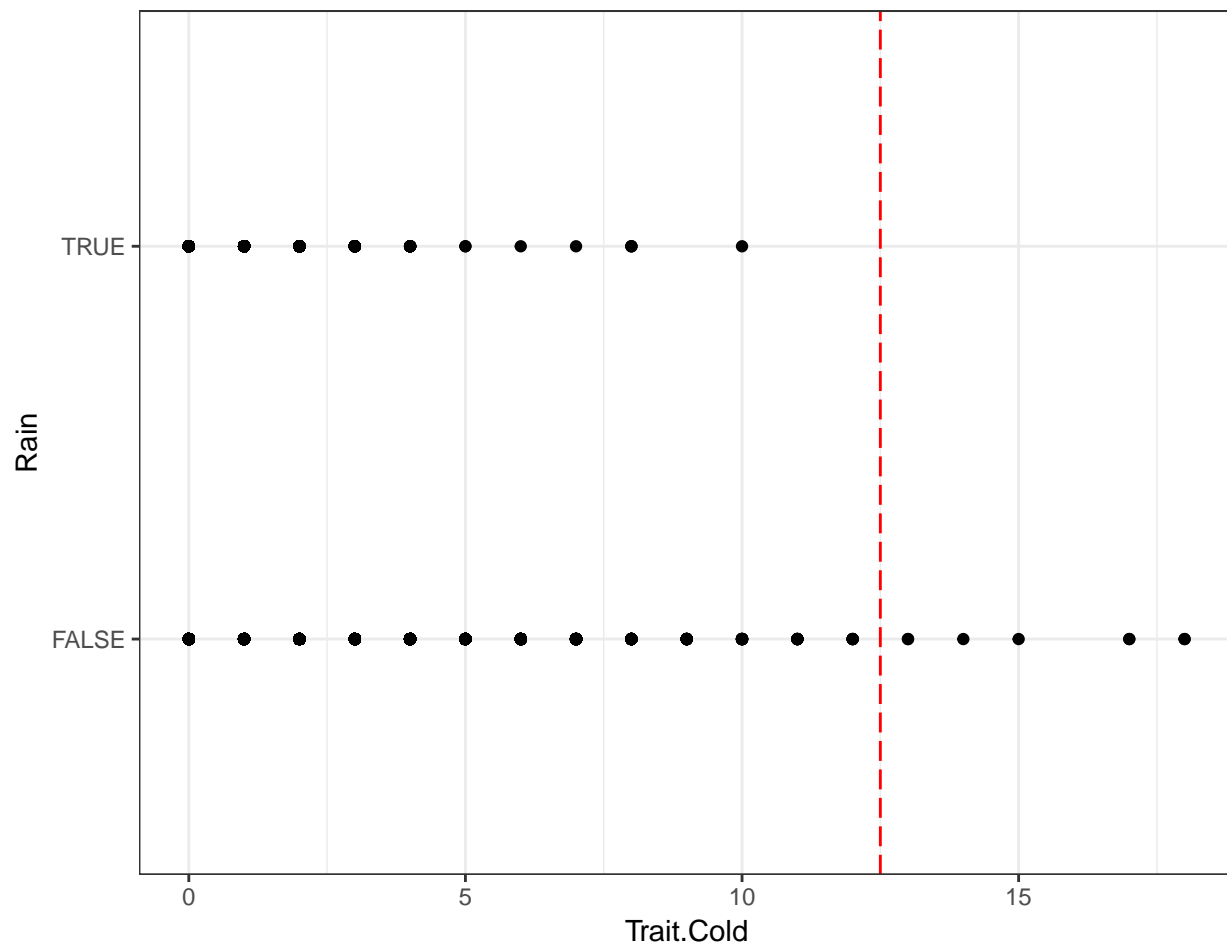




However, the following plot suggests there are some extreme outliers for hours when it did not rain:



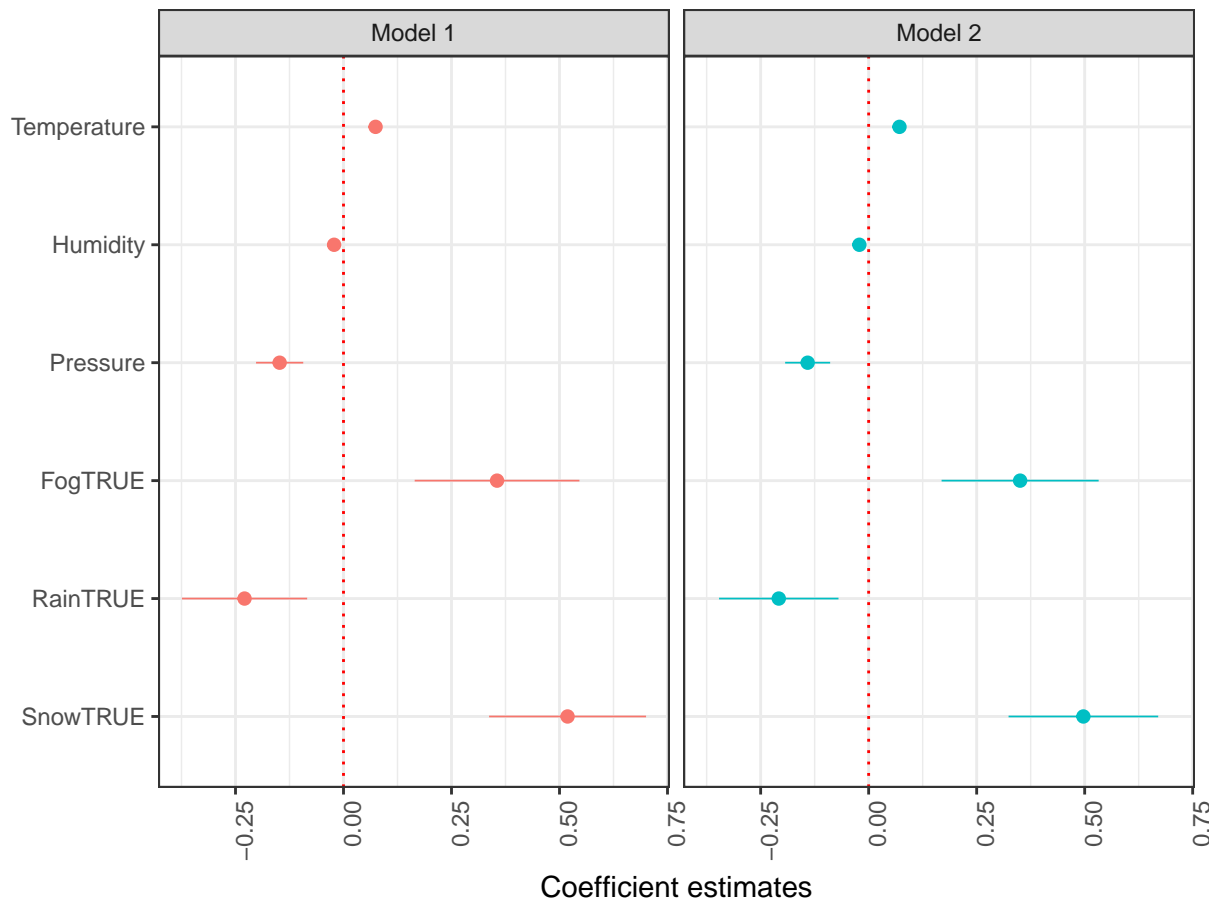
I see four or five extreme outliers. To remove them, I need to find an arbitrary cut-off point.  
The following dot plot confirms that *Trait.Cold*=12.5 can serve as this point:



```
lessOutliers <- dataset %>% filter(Trait.Cold<=12.5)
```

After the outlier observations are taken out, *Rain*'s coefficient estimate remains almost the same:

## Default data vs. no outliers for Rain==FALSE



This suggests that **outliers are not influencing the relationship between *Trait.Cold* and *Rain*.**

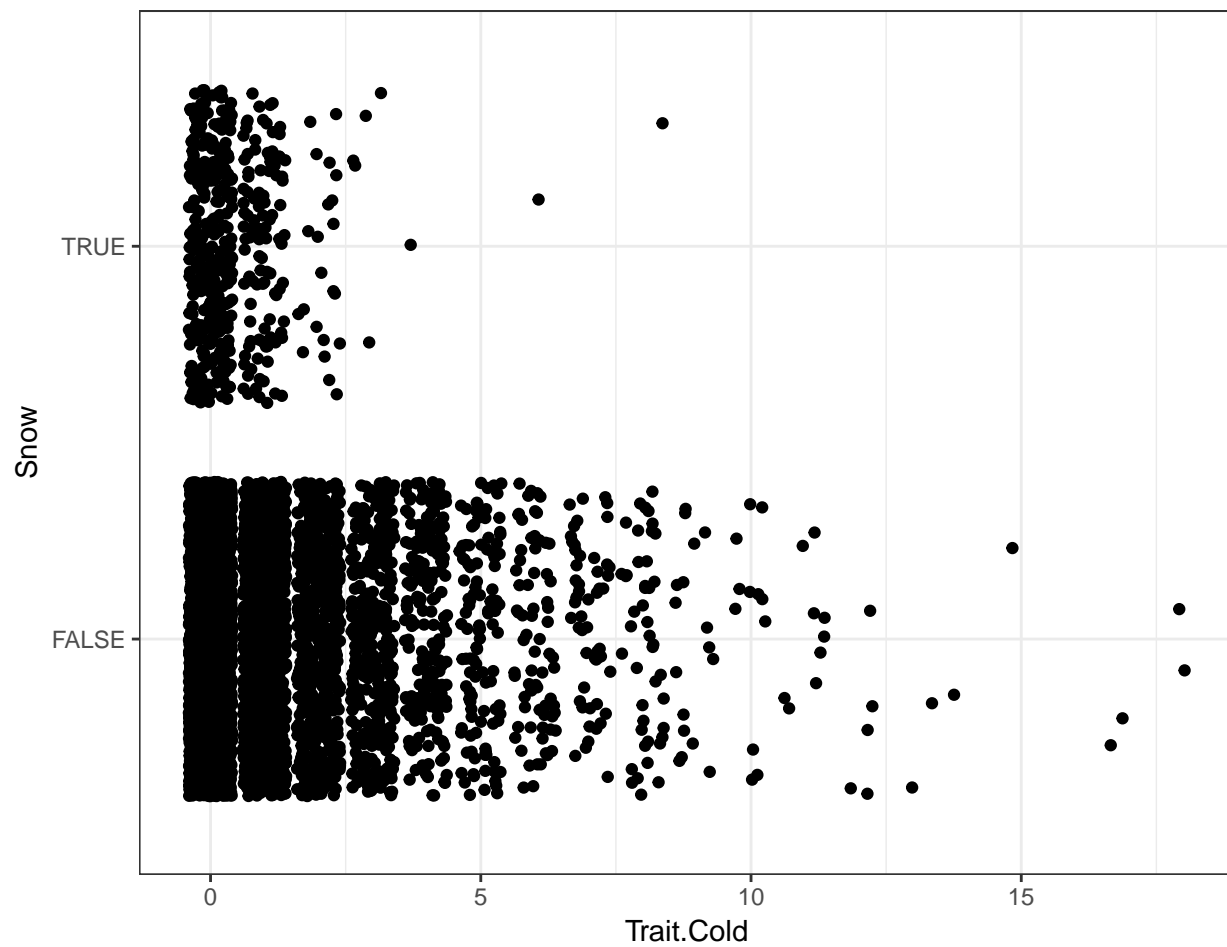
I will repeat the same process for *snow* and *fog*.

## SNOW

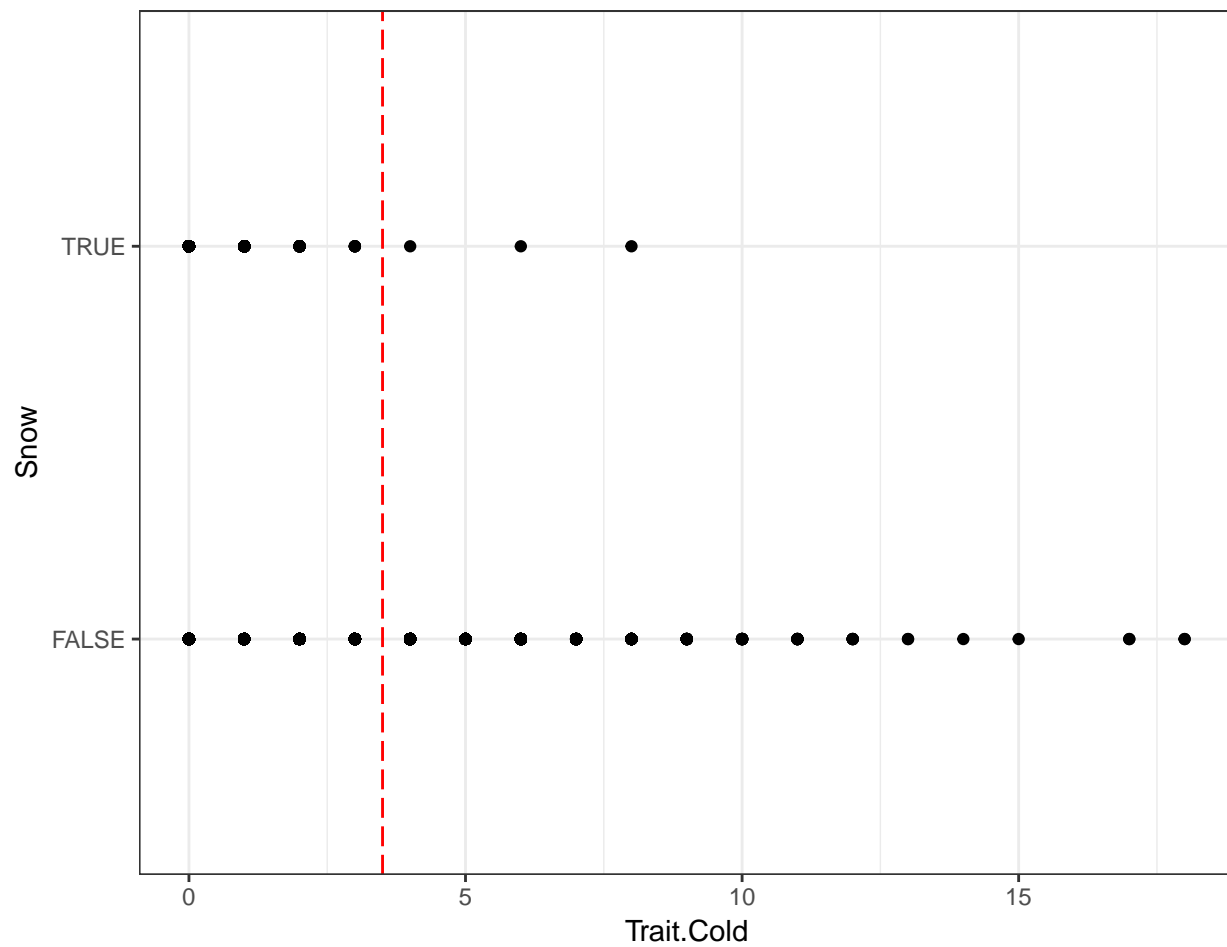
The relationship between *snow* and *cold drink sales* is particularly alarming.

Commonsensically, snow should not be a significant, let alone strong, predictor for sales of cold drinks—even though the data comes from Canada.

There are three outliers for hours when it snowed:



The following point plot suggests that *Trait.Cold*=3.5 is a good cut-off point to capture the three outliers.

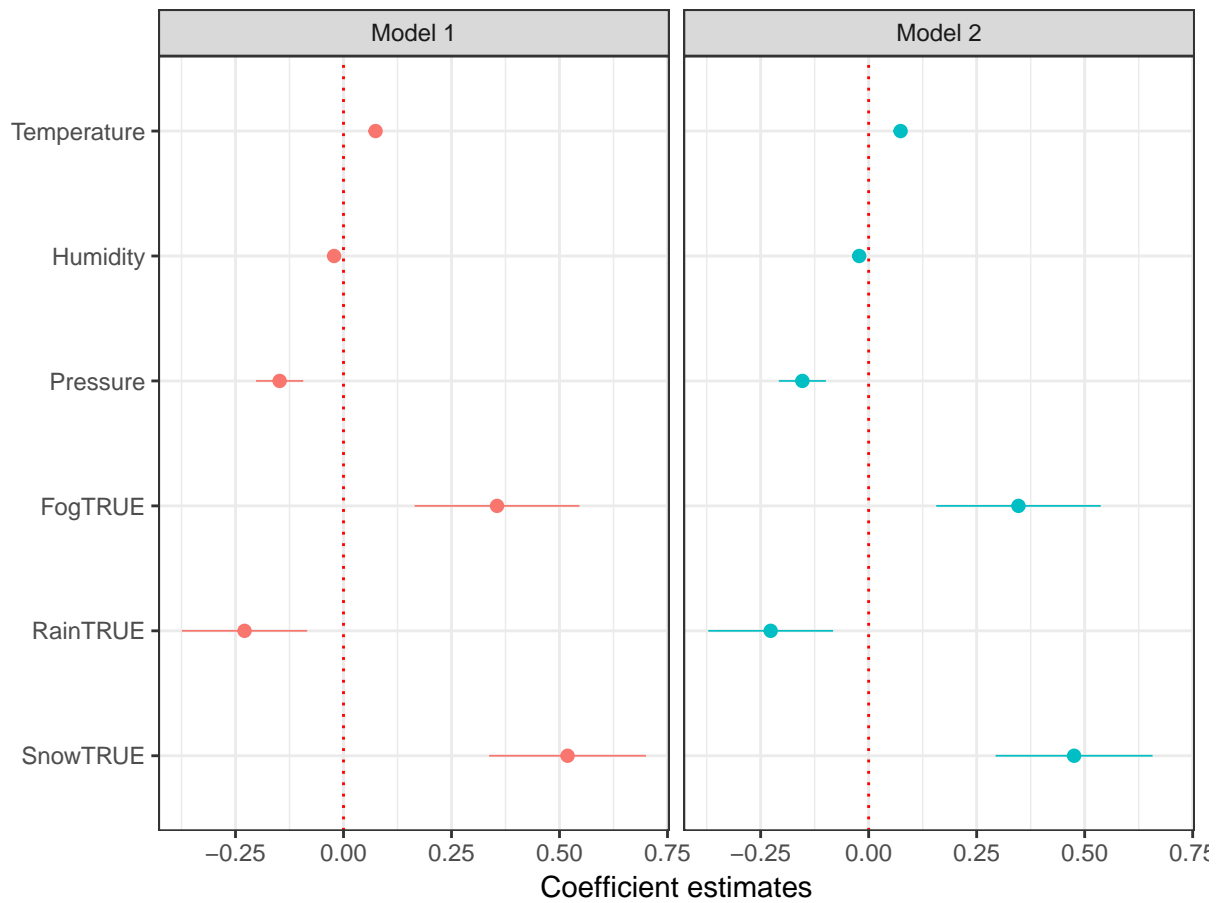


```
outlierSubset <- dataset %>% filter(Snow==TRUE, Trait.Cold >=3.5)
rm(outlierSubset)
lessOutliers <- anti_join(dataset, outlierSubset)
```

After the outlier observations were taken out, *Snow*'s coefficient estimate decreased slightly by around 0.04: far from a drastic change in the relationship with *Trait.Cold*.



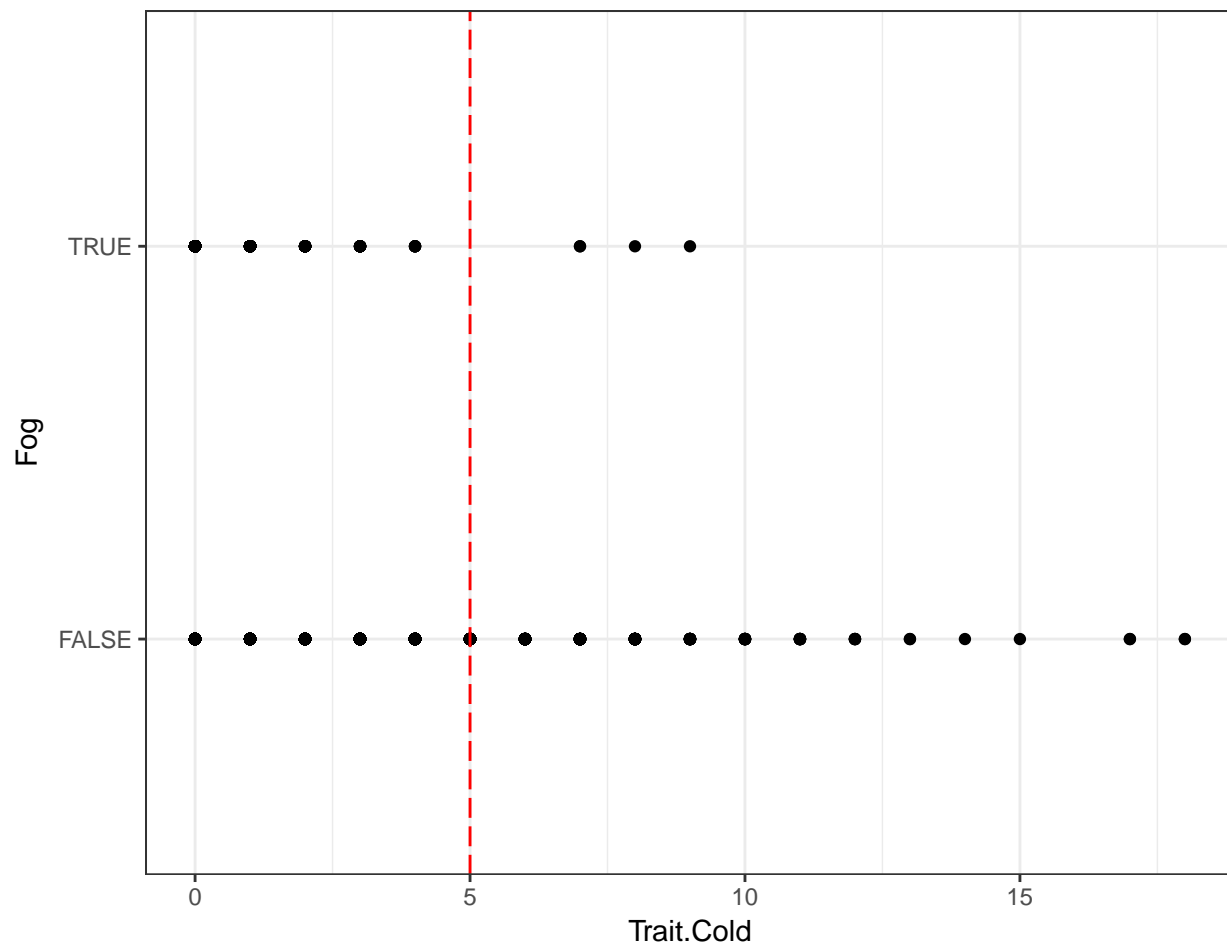
Default data vs. no outliers for Snow==TRUE



This suggests that **outliers are not influencing the relationship between *snow* and *sales of cold drinks*.**

## FOG

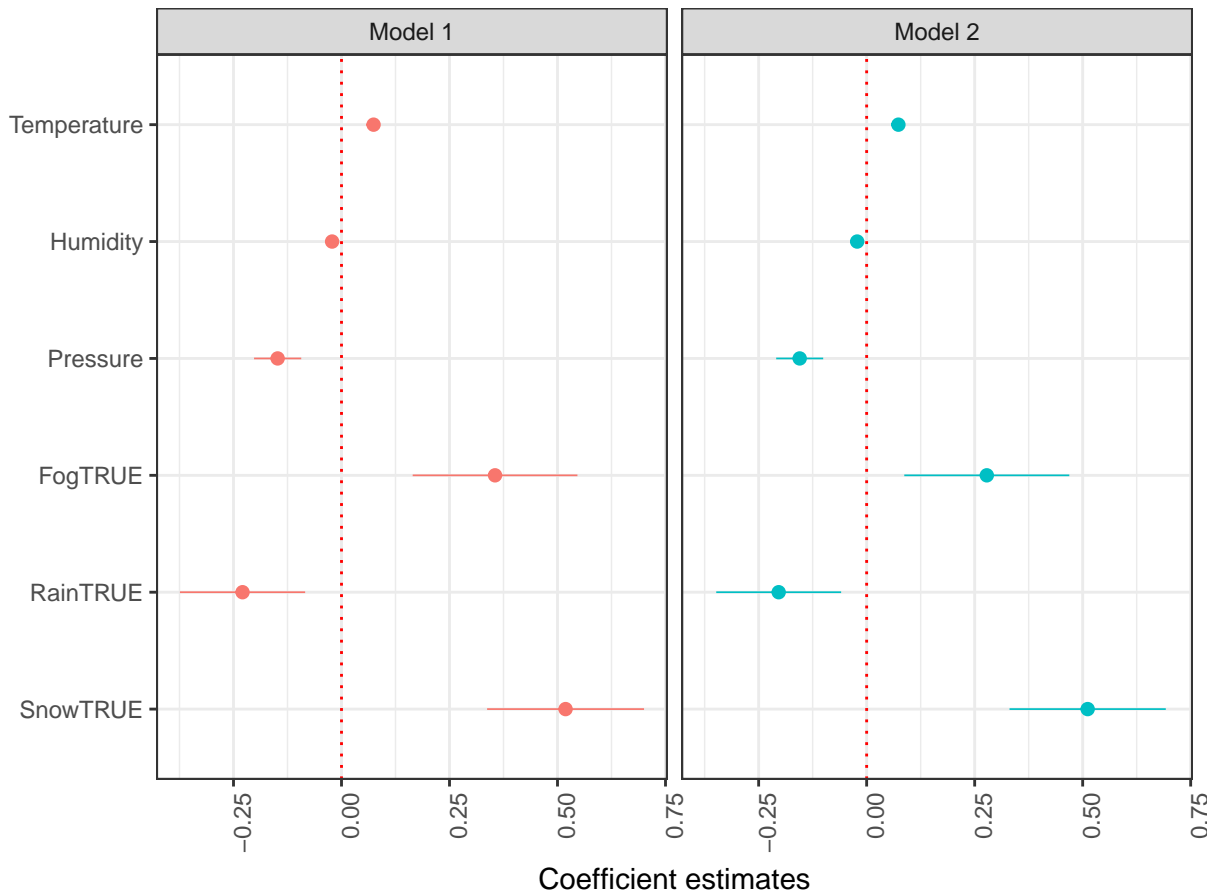
The point plot confirms *Trait.Cold*=5 as the cut-off for four distinct outliers among foggy hours:



```
outlierSubset <- dataset %>% filter(Fog==TRUE, Trait.Cold >=5)
rm(lessOutliers)
lessOutliers <- anti_join(dataset, outlierSubset)
```

*Fog* is still a positive significant predictor for *Trait.Cold*, although its coefficient estimate decreased by around 0.07.

### Default data vs. no outliers for Fog==TRUE



Altogether, removing outliers for each of the binary variables (*rain*, *snow*, and *fog*) changed coefficient estimates neither for the variables for which outliers were removed, nor for other variables in the model.

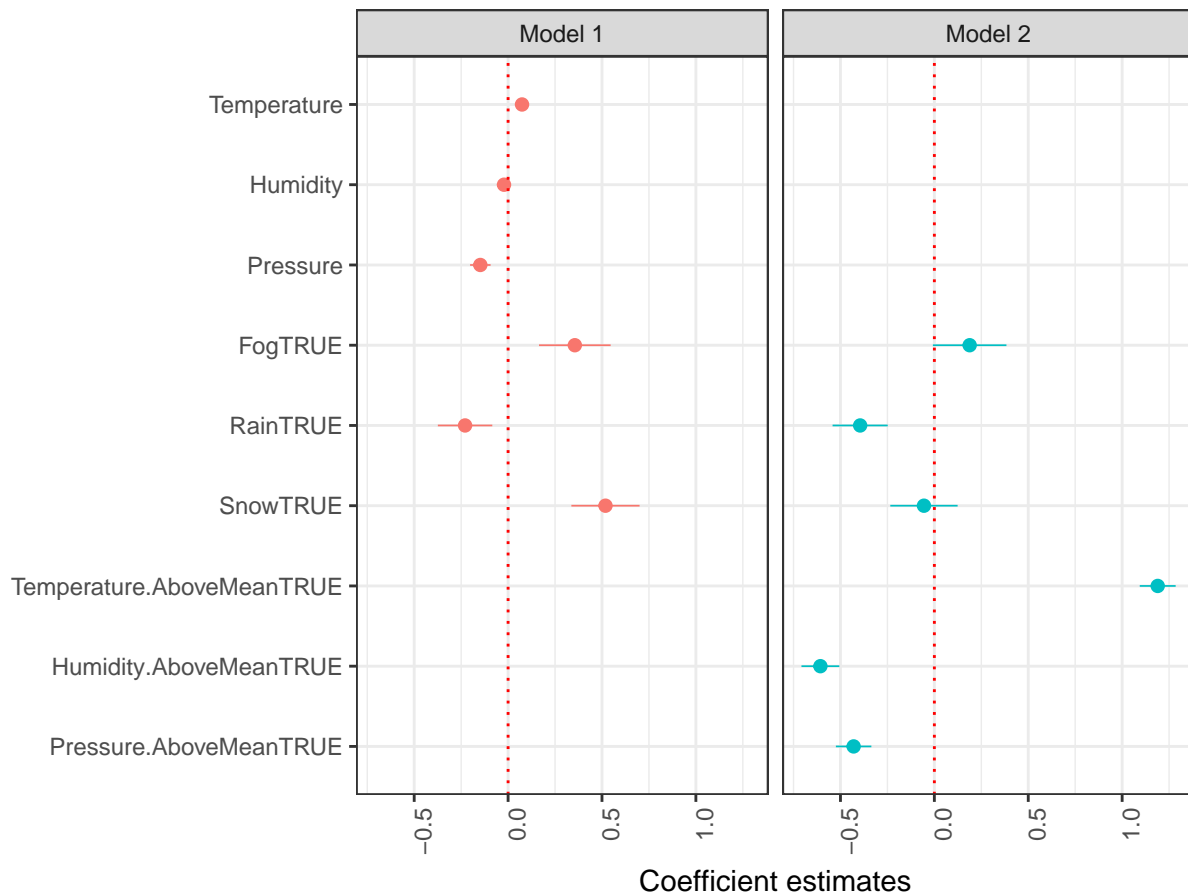
### DIFFERENCE IN PRECISION

The binary variables' comparatively stronger effects on *Trait.Cold*, as measured by coefficient estimates, are not due to extreme outliers.

Earlier, I proposed another cause for this: **the lack of precision in measurement of the binary variables could be what amplifies their effect compared to the continuous variables.**

To see if this is the case, I will restructure the continuous variables, making them binary and run a model with them:

## Default data vs. all-binary data



A significant regression equation was found ( $F(6, 5660)=190.6$ ,  $p < .000$ ), with an  $R^2$  of .17.

With continuous variables recoded as binaries, the situation has reversed:

- Coefficient estimates for *temperature*, *humidity*, and *pressure* moved farther out from zero.
- For the previously binary *fog*, *rain*, and *snow*, the estimates moved closer to zero.

In fact, neither *snow* nor *fog* survived as predictors.

In this preliminary test model with all predictors formatted as binaries,

- *temperature* has a very high coefficient estimate compared to other variables and, together with *pressure*, has the smallest confidence intervals,
- neither *snow* nor *fog* not predict sales of cold drinks, and
- *rain* has a negative effect on sales of cold drinks.

Overall, the results of this model are more intuitive than what I saw previously when modeling using *both* binary and continuous variables.

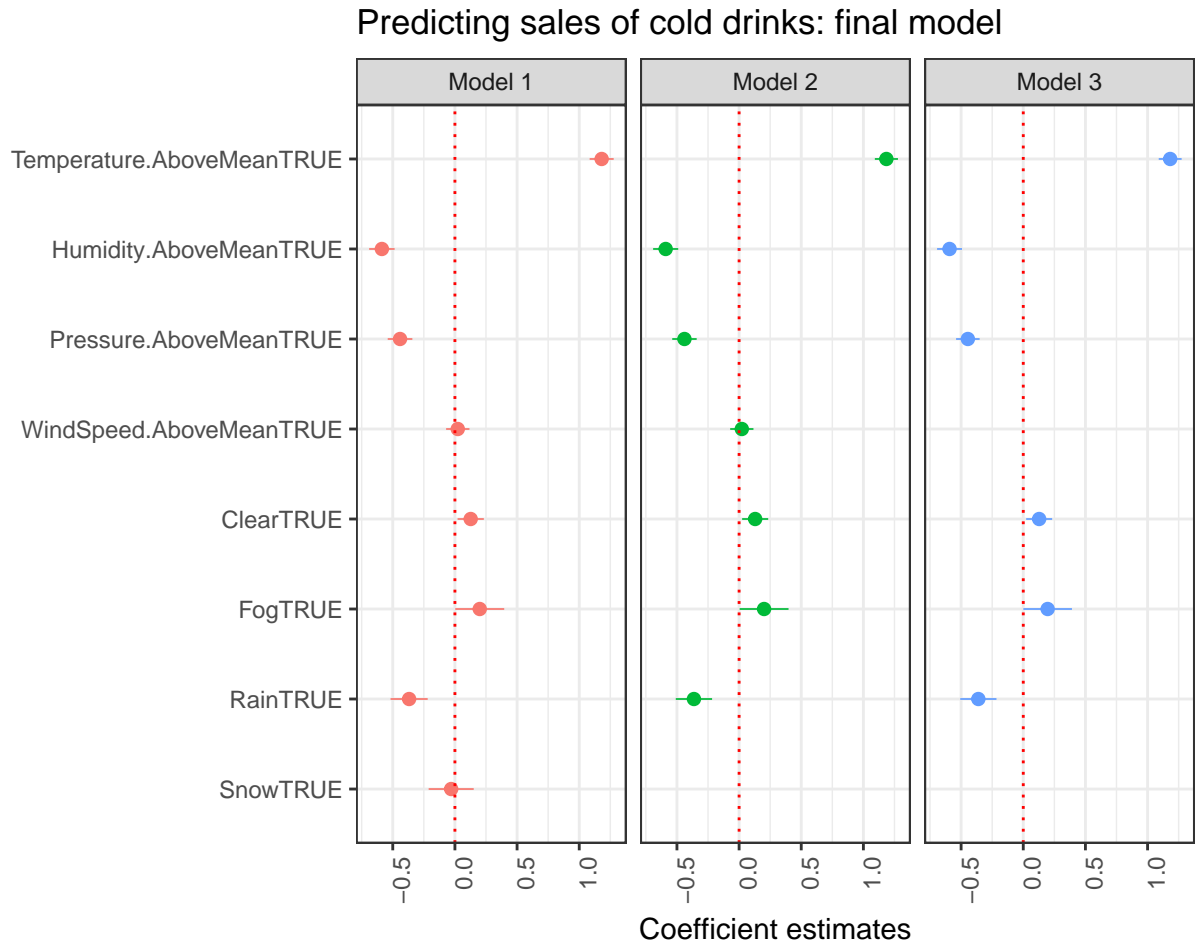
## FINAL MODEL

The above confirms that the lack of precision in measurement of the binary variables amplified the effect of *snow*, *rain*, and *fog* compared to the continuous variables (*temperature*, *humidity*, and *pressure*), since the effects were essentially reversed after converting continuous variables to binaries.

Given this, I will need to create a model explaining *Trait.Cold* from scratch.

This time, the model will have exclusively binary independent variables.

Two iterations of removing insignificant predictors (*snow* first, *wind speed* second) produce the final model:



A significant regression equation was found ( $F(6, 5660)=191.6, p < .000$ ), with an  $R^2$  of .17.

In this final model (model 3 in the above graph), coefficient estimates suggest that

- *above-mean temperature* is a positive predictor for sales of cold drinks, with an effect farthest from zero compared to the other variables,
- *above-mean humidity* follows as a negative predictor,
- *above-mean pressure* and *rain* are negative predictors, both roughly at the same level, and
- both *clear sky* and *fog* are somewhat significant\* positive predictors, with their effects closest to zero compared to the other variables.

Judging by these results, I will be able to confidently report on the effects of temperature, humidity, pressure, and rain on sales of cold drinks.

Visibility, on the other hand, does not appear to make a difference for sales of cold drinks. Clear skies and fog are less significant than the other variables. Plus, they signify opposite concepts, and hence cannot both be positive predictors.

**Regarding the hypothesis for *Trait.Cold*, I can conclude that**

Higher numbers of cold drinks sold coincide with above-average temperatures.

However, above-average pressure and humidity are both associated with lower numbers of cold drinks sold.

Finally, visibility, measured by the presence of clear sky or fog, does not appear to be a factor in sales of cold drinks.

## RECOMMENDATIONS

Based on my results, I would recommend my client to:

1. **Create a weather trigger-based marketing strategy for cold drinks.** In addition to above-average temperatures, the trigger should also include above-average humidity and pressure, which both coincide with lower sales of cold drinks. A trigger-based marketing strategy could help offset the negative effects of the two.
2. **Use in-store displays to emphasize that cold drink options are available both “to go” and “for here.”** Higher sales of cold drinks do not coincide with the presence of clear skies, and neither does high pressure—generally *associated* with good weather. So, the client should not assume that cold drinks are only suited for weather that lends itself nicely to walking around with an iced latte in hand.

## FUTURE RESEARCH

I will continue analyzing the remaining dependent variables in my dataset.

I am particularly interested in finding out whether sales of espresso- and tea-based drinks coincide with different weather conditions.

Do people opt for espresso-based drinks in certain weather conditions, but not in others? Would there be enough evidence to suggest that in weather conditions unfavorable for espresso, sales of tea-based beverages can compensate for a dip in sales in espresso-based drinks?

These are all exciting questions!

There are also opportunities to explore the dataset beyond the effects of weather variables.

For example, take milk—a particularly expensive part of every drink, compared to other ingredients.

The client may find that they can’t quite put their finger on how many milk cartons to order—sometimes a few goes to waste, or, instead, there’s always not enough.

Using the dataset, I may uncover hourly, daily, or weekly trends in sales of items containing milk and suggest ways to save money through smarter stock management.