Term Paper

BAN440 – Data Driven Business Analysis

NHH

Balancing Demand and Social Mandates: A Data-Driven Approach for Vinmonopolet's Store Establishments in Norway

Candidate numbers: 74, 79, 85

Date: 11.04.25 – 25.04.2025

Pages: 10 (excl. front page, table of contents and references)

Contents

Introduction

The Norwegian alcohol retail market is uniquely regulated through a state-controlled monopoly, markedly different from typical consumer sectors. Vinmonopolet, the exclusive retailer of beverages with alcohol content above 4.75%, operates under a dual mandate: safeguarding public health and ensuring equitable access.

To guide its expansion decisions, Vinmonopolet has publicly outlined a set of store establishment criteria. These include, but are not limited to, population size, purchasing power, tourism activity, distance to the nearest outlet, and the overall economic viability of a potential store (Vinmonopolet, 2025b). The guidelines provide not only operational direction but also a rare opportunity to analyse how such principles can be translated into a formal, data-driven model to inform decision making.

Following this, our term paper aims to investigate whether it is possible to use publicly available data to replicate or approximate Vinmonopolet's internal store establishment decisions. Specifically, we ask:

"Using publicly available data, can we construct a data-driven model that could assist Vinmonopolet in their store establishment decisions for new markets?"

Entry decisions in regulated monopolies, particularly those where the monopolist has social mandates, pose distinct analytical challenges. Vinmonopolet's decisions are not purely profit-driven but reflect both demand-based viability and equitable access to public goods. Our empirical approach is designed to reflect these priorities. By operationalizing Vinmonopolet's establishment criteria, we attempt to construct a model that combines predicted demand with observed socioeconomic characteristics to estimate the probability of store presence in municipalities without a store.

Literature review

The economic literature on firm entry and market structure has long emphasized the role of observable market characteristics in shaping market presence decisions. In their influential work, Bresnahan and Reiss (1991) developed a framework for estimating how variables such as market size and local demand conditions affect the number of active firms in a given market. Although their focus lies primarily on competitive environments, the conceptual structure remains relevant in contexts where decisions about market presence are made under institutional or regulatory constraints.

Subsequent contributions, such as Berry (1992), expanded on these ideas by applying discrete choice models to firm entry, demonstrating how market-level heterogeneity can be incorporated into probabilistic modelling frameworks. More broadly, Einav and Levin (2010) underscore the value of flexible empirical strategies that integrate multiple econometric tools, especially when modelling firm behaviour in diverse industry settings. Their emphasis on methodological adaptability provides a useful foundation for selecting models that reflect both the structure of the data and the institutional setting of the application.

Complementing this emphasis on flexibility, recent studies have also shown that traditional methods like ordinary least squares (OLS) may still be well suited to smaller datasets. Boef et

al. (2014) demonstrate that OLS may outperform instrumental variable (IV) methods in practical applications, particularly when instruments are weak, or confounding is minimal. This underscores the value of methodological simplicity and clarity under data constraints, especially when the objective is prediction rather than causal inference.

A description of the Norwegian alcohol retail market

As previously mentioned, Vinmonopolet is the only entity legally permitted to sell alcoholic beverages with an alcohol content exceeding 4.75% in retail stores. Consequently, it bears a significant social responsibility regarding the placement of its stores, the selection of products offered, and the overall accessibility of these outlets (Vinmonopolet, 2025a). In the pursuit of establishing sound norms for this process, Vinmonopolet has outlined a set of criteria for determining suitable store locations. Broadly speaking, these criteria encompass the goal of ensuring that each store is financially viable on its own, as well as promoting accessibility to the public, with the underlying principle that the majority of the population should have reasonable access to a Vinmonopolet outlet (Vinmonopolet, 2025b).

Financial viability is determined by using characteristics such as purchasing power, population density, distance to the nearest Vinmonopolet store, the presence of urban centres or shopping hubs within the municipality, and various situational criteria, including tourism and cross-border trade. Considerations of accessibility include assessing how far individuals must travel to reach the nearest Vinmonopolet outlet(Vinmonopolet, 2025c).

If a municipality wishes to establish an outlet, it must apply to Vinmonopolet centrally. All municipalities that have applied are granted *active* status, meaning they are eligible for consideration when determining the establishment of new stores. Municipalities that have not applied will not be considered. It is sufficient for a municipality to apply once to maintain active status. If Vinmonopolet's analysis concludes with establishing a new outlet in a municipality, a process is initiated to identify a suitable location and premises for a store.

Due to Norway's vast and sparsely populated areas, it is often a matter of judgment whether to establish stores in locations that may not even break even, to prevent certain segments of the population from having exceptionally long distances to the nearest Vinmonopolet store. An attempt to replicate this decision process using publicly available data is the aim of this report.

An overview of our data

Our dataset combines multiple publicly available sources to approximate Vinmonopolet's establishment criteria. Store-level data on locations, operational status, and annual sales volumes were obtained from Vinmonopolet's open API. Demographic, geographic, and socioeconomic variables such as population, income, tourism activity, and land area, were sourced from Statistics Norway (Aasestad, 2025; Bye et al., 2025; Haug, 2025; Høydahl, 2025; Rapp & Bloch, 2024), while geospatial data used to calculate aerial distances was retrieved from the Norwegian Mapping Authority (Kartverket, 2024, 2025).

Variables

In table 1 below, we briefly describe explanatory variables that were considered in our models. Although other variables were considered, only the retained variables are displayed.

Variable	Description	Source
number_of_stores	The number of stores in the municipality	Vinmonopolet API
sales	Annual alcohol sales per municipality (in thousand liters)	Vinmonopolet
population	Total inhabitants aged 18 and over in each municipality (in thousands)	SSB
grensehandel	Measured as estimated alcohol spending in million NOK per municipality	SSB
n_stays	Number of overnight stays per municipality per year (in thousands)	SSB
monthly_salary	Average gross monthly earnings (in thousand NOK) per inhabitant in the municipality	SSB
dist_nearest	Aerial distance in kilometers from the administrative center of each municipality to the closest Vinmonopolet store	Kartverket/SSB
active	Whether a municipality has applied for having a Vinmonopolet store	Vinmonopolet

Table 1 - Key variable descriptions

These variables are designed to align with Vinmonopolet's establishment criteria, but doing so requires several non-trivial modelling and transformation choices, further outlined below.

Operationalizing establishment criteria

Vinmonopolet's establishment guidelines emphasize several broad factors. Translating these qualitative criteria into measurable variables required a series of pragmatic modelling choices aimed at balancing parsimony, interpretability, and relevance to a real-world decision context.

Certain variables were straightforward—for instance, population is restricted to inhabitants aged 18 and over, in line with legal purchasing eligibility. Others required more interpretation. Tourism activity (*n_stays*) is based on the number of overnight stays in hotels, cabins, and similar accommodations, though coverage is incomplete due to inconsistent municipal reporting. Cross-border shopping (*grensehandel*) is estimated from regional-level data and allocated to municipalities weighted on population.

Accessibility was proxied using aerial distance (*dist_nearest*) from each municipal's administrative centre to the nearest Vinmonopolet store, as road-based data proved harder to obtain. While this simplification may understate travel distances in some areas, it provides a consistent geographic baseline. Purchasing power was captured through gross average monthly salary (*monthly_salary*) at the municipal level.

Data handling and Limitations

Constructing the dataset also presented practical challenges. Some variables, such as overnight stays (*n_stays*), suffer from missing values due to inconsistent reporting across municipalities, especially among smaller ones. In other cases, cross-border shopping data (*grensehandel*) was only available at the regional level and had to be scaled down to municipality level. This approximation introduces potential measurement errors but could still capture spatial variation in cross-border shopping effects.

Another challenge stemmed from recent territorial reforms in Norway, where several municipalities have been merged or split following political decisions. Ensuring consistent coding and alignment across all datasets required manually harmonizing municipality codes to the 2024 structure, a task that involved both administrative and geographic reconciliation.

These issues underscore the importance of a critical lens when interpreting our model results. While our modelling strategy emphasizes simplicity and transparency, we recognize that underlying data limitations may obscure more subtle patterns. Nonetheless, we view this as a reasonable and replicable approximation of the criteria Vinmonopolet uses in practice, and one that offers valuable insights even under constrained data conditions.

Exploratory Data Analysis

To better understand the structure of our dataset and guide subsequent modelling decisions, we begin by visualising key characteristics of the municipalities. Figure 1 displays the number of Vinmonopolet stores by municipality across Norway. Most municipalities have either no store ($n = 0$) or just one ($n = 1$). In the relative scale of our dataset, only a small number of municipalities host more than two stores, for example Oslo (with 34) and Bergen (with 11).

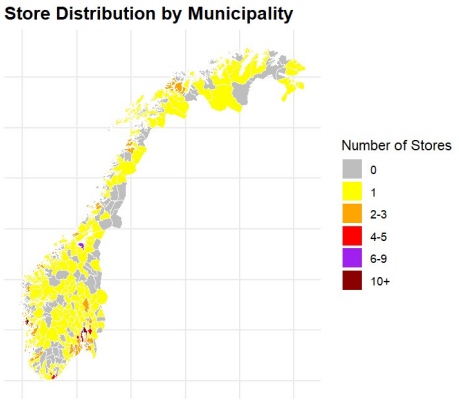


Figure 1: Map of store distribution by Municipality

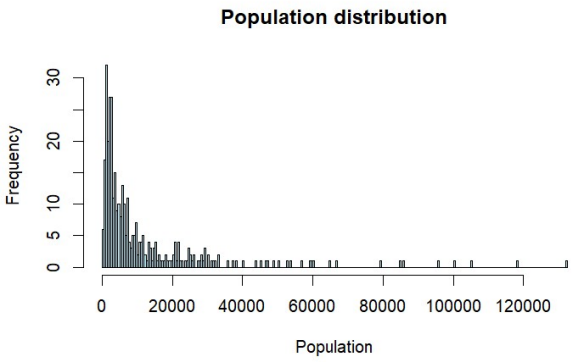


Figure 2: Population distribution by municipalities without the four biggest cities.

Figure 2 presents the distribution of municipal population sizes. As shown, the population data is highly skewed, with most municipalities having fewer than 20,000 residents and only a few significantly exceeding that number. These high-population municipalities like Oslo and Bergen are not only statistical outliers but could also be qualitatively different from smaller municipalities in terms of infrastructure, population density, and service provision. This skewness raises a concern that the large municipalities might exert disproportionate influence on model estimates if included without adjustment.

Together, these plots suggest that *population* is a likely key driver of store placement and that municipalities with moderate to low population levels form the core of the decision-making landscape for Vinmonopolet. They also highlight the diversity and imbalance in Norway's municipal structure, an important contextual feature that our model should consider.

Empirical strategy and models

Our end goal is to create a model that will estimate how "attractive" it is for Vinmonopolet to open a store in a municipality that does not currently have one. Subsequently our approach

aims to estimate a binomial logit model that gives us the probability of store presence in each municipality. Hopefully, the estimation will align closely with Vinmonopolet's own decisions.

At first glance, this approach appears intuitive, but there's a fundamental issue: municipalities without stores lack sales data. If we intend to use sales figures to estimate the likelihood of a store's presence, we first need a measure of local demand. By leveraging sales data from municipalities with stores, we can estimate demand levels for those without, effectively transforming the problem into a two-step estimation process.

Model choices

In our analysis, we selected a standard linear regression model to estimate demand, drawing from Boef and colleagues (2014) which highlights the advantages of Ordinary Least Squares (OLS) for its transparency and interpretability in estimating continuous outcomes such as sales volumes. This choice is particularly relevant for municipalities lacking an existing retail presence, aligning with Vinmonopolet's strategic focus areas, such as local demand drivers.

For predicting the probability of a municipality hosting a store, drawing from Berry (1992), we employ a binomial logit model, This decision underscores the model's ability to handle the non-linearity inherent in binary outcomes, ensuring predicted probabilities remain within appropriate bounds. The two-step approach is informed by empirical frameworks established by Einar and Lewin (2010), in this context applied to entry decisions in regulated monopoly settings. We aim to effectively adapt these frameworks to the unique institutional and data-specific characteristics of the Norwegian alcohol retail market in the modelling approach.

Given our relatively small dataset, we have drawn from Boef and colleagues (2014), avoiding the use of more complex machine-learning methods such as random forests or gradient boosting. Additionally, in line with Bresnahan and Reiss (1991), we focus on broad, observable market characteristics to serve as proxies for establishment criteria, reflecting a preference for models that are transparent, replicable, and grounded in publicly available data.

Step 1: Demand estimation with linear regression

Before estimating demand, we needed to consider model selection. Both forward and backwards stepwise variable selection removes *Spread* from the regression. We also want to remove *Area* as it is not statistically significant in predicting sales for the smaller municipalities in Norway. *number_of_stores* is also removed as the analysis revolves around municipalities without a store. This leaves us with the following linear regression:

$$sales = \beta_0 + \beta_1 \times population + \beta_2 \times grensehandel + \beta_3 \times n\_stays + \beta_4 \times monthly\_salary + \varepsilon$$

After the OLS estimation, we should be left with predicted sales data for the municipalities without a store. We encountered an issue where the negative regression coefficients together with the intercept accumulated negative sales volumes for some municipalities. In the further analysis we have decided to recode the negative values as 0. This is of course a weakness of estimating a linear model as one would never expect negative, or even close to zero, in sales volume. Nevertheless, municipalities that get negative estimates from our model will likely not get any good results with a model that prohibits this from happening either. Since our

focus is on opening new stores, we are primarily interested in municipalities where sales are predicted to be high, rather than those with lower sales forecasts.

Step 2: Discrete choice with binomial logit

Our data now has complete information in terms of estimated sales volumes for all municipalities. This becomes useful when predicting the probability of there being a store in municipalities without one currently. The probability that the number of stores is k given the characteristics X_i:

$$P(Y_i = k|X_i)$$

The linear predictor V_i can be specified as:

$$V_i = \beta_0 + \beta_1 \times sales + \varepsilon$$

The sales variable used here is itself the product of an earlier regression model, where it was estimated using multiple observable characteristics. Including those same variables again in the logit step would reintroduce them indirectly, increasing the risk of overfitting. Since our focus is on municipalities without a store, where sales had to be estimated, these form the natural basis for prediction. To avoid redundancy, we use only predicted sales as the input for store presence. Model performance is validated using 5-fold cross-validation, which shows that this simpler specification slightly outperforms the more complex alternative in terms of ROC score, indicating better generalizability.

The probability of having a store $(Y_i = 1)$ is modelled using the logistic function:

$$P(Y_i = 1|X_i) = \frac{1}{1 + e^{-V_i}}$$

where Y_i is the binary outcome variable (1 if there is a store, 0 otherwise), X_i represents the vector of independent variables for observation i. β_i are the coefficients to be estimated, e is the base of the natural logarithm, and the error term ε is implicitly included in the logistic regression model as part of the stochastic component of the binary outcome.

Distance to the nearest store (*dist_nearest*) measures the straight-line (aerial) distance from a municipality's administrative centre to the closest Vinmonopolet store. Given Norway's rugged terrain and winding roads, aerial distance may understate actual travel distance to a store, particularly considering Vinmonopolet's stated accessibility goal. While the variable is clearly policy-relevant, we opted not to include *dist_nearest* as a predictor. This was shaped by both practical modelling considerations and alignment with establishment criteria. In municipalities that already have a store, the distance to the nearest outlet is small, often close to zero, making it uninformative in the training data. In contrast, distance in non-store municipalities, by definition, is larger, and with more variability. This asymmetry risks introducing a misleading negative relationship between distance and store presence: the model might learn that being far from a store reduces the likelihood of having one, which runs counter to the institution's stated aim of improving access in underserved areas.

Additionally, there is little to suggest that Vinmonopolet applies a strict distance cutoff; rather, decisions are likely shaped by case-by-case judgment. Considering this, we found it more appropriate to treat *dist_nearest* as a contextual lens in our interpretation of the model results.

This way it can be used as something to enrich our understanding of the results, even if not directly modelled. We argue that this preserves statistical coherence while remaining sensitive to the broader social and geographic realities that influence store placement decisions.

To summarize, our model could output reasonable probability estimates to help decision making, but distance needs to be discussed separately.

Model results

Demand - Sales

The regression output indicates that population size is by far the strongest (and most precisely estimated) driver of annual sales, with each additional resident increasing sales by roughly 16 litres annually. *grensehandel* has a negative relationship with local sales—every thousand NOK in cross-border spending each year is associated with about 5 fewer litres sold domestically. Tourism also boosts local sales, though more modestly; each additional overnight stay raises expected sales by roughly 0.25 litres. Finally, a higher average monthly salary is associated with an increase in annual sales. A one-thousand-NOK increment in the monthly average results in an annual increase in sales of 5 000 litres, suggesting that purchasing power plays a meaningful role in driving demand. Overall, these four variables collectively explain nearly all variation in our data ($R^2 = 0.99$). One possible reason for this is that all municipalities with one or more stores are included in the analysis. As a result, the largest cities, being significantly larger than the average municipality, dominate the dataset. We can reaffirm this statement by running a simple linear regression model with *population* as the only independent variable and seeing that it explains 98.66% of *sales*. Conversely, when running the same regression on only municipalities with 0 or 1 stores the R^2 is 71.21%, suggesting that other factors also could be of significance. With this variable as our best alternative for market size, we choose to continue with a model with such a large R^2.

Regression output from demand estimation

Variables	*sales*		Standard deviation
population	16.449	***	0.490
grensehandel	-5.264	***	1.456
n_stays	0.246	***	0.043
monthly_salary	4.885	**	2.290
Constant	-270.085	**	125.336
Observations	237		
R2	0.990		
Adjusted R2	0.990		
Residual Std. Error	102.638		
F Statistic	5,683.347	***	

Table 2 - Regression output from demand estimation

Ideally, we would restrict the training data to municipalities with exactly one store, as this might provide a more balanced representation. However, when doing so, we observe that the model becomes unrepresentative at the national level, as certain regions become underrepresented. As an example, the effect of *grensehandel* disappears completely with a restrictive data set, neglecting the large effect we observe this to have in the areas close to the Swedish border, like the region Østfold. This happens because there are few municipalities in this region, and that filtering down to one store will remove such a large proportion of the population and municipalities of the region that the effect disappears. We therefore choose to include all municipalities with at least one store to maintain variable significance, even though we acknowledge this leads to an overrepresentation of urban areas.

Discrete choice - Probabilities

Table 2 ranks the 10 highest scoring municipalities for establishing a new store according to our model, which is based on predicted sales. The table also exhibits that the model favours *population*, as many of the high-ranking municipalities have a large population compared to other municipalities without a store.

mun_name	prob	dist_nearest	sales	population	region_name	active
giske	0.995	3	156	8.773	Møre og Romsdal	0
lunner	0.993	7	150	9.420	Akershus	1
rade	0.967	13	118	7.850	Østfold	0
hareid	0.857	10	87	5.320	Møre og Romsdal	1
valer (ostfold)	0.823	11	82	6.162	Østfold	1
aurland	0.817	37	81	1.836	Vestland	1
birkenes	0.752	13	74	5.413	Agder	0
eidskog	0.729	23	71	6.059	Innlandet	1
aure	0.683	28	67	3.394	Møre og Romsdal	1
austrheim	0.658	16	65	2.915	Vestland	1

Table 3 - Ranking the 10 best "0" municipalities according to our model

As discussed, it is necessary for a municipality to apply for a store to be considered for one. To mimic Vinmonopolet's decision making process, we filter out the municipalities that have not applied and display our model's 10 best candidates that have applied in table 3 below.

mun_name	prob	dist_nearest	sales	population	region_name	active
lunner	0.993	7	150	9.420	Akershus	1
hareid	0.857	10	87	5.320	Møre og Romsdal	1
valer (ostfold)	0.823	11	82	6.162	Østfold	1
aurland	0.817	37	81	1.836	Vestland	1
eidskog	0.729	23	71	6.059	Innlandet	1
aure	0.683	28	67	3.394	Møre og Romsdal	1
austrheim	0.658	16	65	2.915	Vestland	1
aukra	0.644	15	64	3.759	Møre og Romsdal	1
vaksdal	0.617	19	61	3.875	Vestland	1
sokndal	0.544	21	55	3.371	Rogaland	1

Table 5 - Ranking the 10 best active "0" municipalities according to our model

Figure 3 displays output from the logit model in terms of location and probability of having a store in municipalities without one, coloured in increments of 25%. We observe the high-ranked municipalities to be sporadically distributed across the country, while the lower ranked ones seem to be in low population areas, close to the Swedish border. In the following part, we conduct a contextual analysis of model output and discuss robustness.

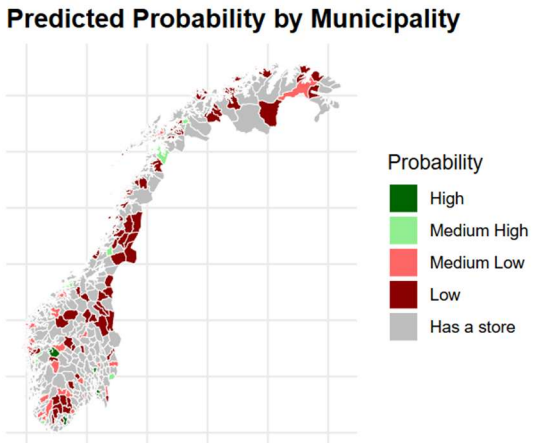


Figure 3 - Predicted probability in "0" municipalities

Tourism

We observe that 7 out of the 10 best municipalities have missing *n_stays* values. This may indicate that the variable is not particularly important unless the municipality has substantial tourism. As mentioned, there are large amounts of missing data for *n_stays*, meaning that our model could have missed out on attractive prospects for store establishment. Nevertheless, based on our predictions, there

are no clear cases of municipalities with missing values for *n_stays* that are likely to have substantial tourism, given our limited knowledge of Norway's tourism.

However, we observe that Aurland is among the top candidates on the list, despite having significantly fewer inhabitants than the others. Thus, high tourist activity likely contributes to the high estimated probability that a Vinmonopolet store should be located there. Interestingly, Vinmonopolet has in fact announced that it plans to open a store in Aurland during 2025.

Cross-border shopping

The real-world values for *grensehandel* could vary greatly just based on proximity to the Swedish border. One would expect the cross-border shopping to be larger in the municipalities closest to the border. This could likely be the case for Eidskog, a municipality that has a direct border to Sweden, but did well in our model.

The omitted-distance bias

When comparing the tables of top-ranked municipalities, one notable pattern emerges: those assigned the highest predicted probabilities of receiving a store tend to be located relatively close to an existing Vinmonopolet outlet. This is unsurprising, as our model does not account for proximity to current stores, as it lacks a distance-based penalty/reward. As a result, this is reflected in municipalities such as Giske and Hareid, which are within short travel distances of existing outlets, and receive some of the highest scores. Interestingly, Giske has not even applied for a store, pointing to a disconnect between model outputs and real-world decisions.

Meanwhile, we observe that remote municipalities like Karasjok (67 km) and Tana (60 km) , arguably facing some of the greatest accessibility challenges, receive comparatively low predicted probabilities. This example also reflects the absence of *dist_nearest* from our model. Without a mechanism to favour underserved areas, the model naturally prioritizes demand-based predictors, which may not fully capture Vinmonopolet's broader equity objectives.

Interestingly, among the top-ranked municipalities without a store, Aurland stands out not only for its high model score but also for its considerable distance to the nearest outlet, over 35 kilometers. As noted earlier, Vinmonopolet has confirmed plans to open a store there in 2025. This suggests that distance, while excluded from the predictive model, plays an important role in practice. We argue that incorporating distance formally would further enhance the model's ability to reflect Vinmonopolet's accessibility goals.

Why haven't all municipalities applied?

It is not entirely clear why some high-scoring municipalities have not applied for a Vinmonopolet store. Local political considerations, differing priorities, or other contextual factors may play a role, many of which fall outside the scope of what can be captured through our dataset. Additionally, there may be slight discrepancies in the data on application status. For instance, Råde is reported to be planning a new store despite not appearing as active on Vinmonopolet's public list, likely due to the timing of application updates (Lunheim, 2024).

Local ad hoc effects to note

There are also local factors that influence both the decision to apply for a Vinmonopolet outlet and the decision to approve such applications. Typical examples include a lack of drinking culture, often associated with strongly religious municipalities, or high levels of through-

traffic, which may result in the population figures not accurately reflecting the number of people present in the municipality. Municipalities that host major attractions or events without this being reflected in overnight stay statistics fall into the same category. The common denominator among these cases is that, while they do not represent a major driver in determining whether a municipality *should* have a Vinmonopolet outlet in general, they can be decisive in specific cases where such local conditions are present.

Limitations and suggestions for future research

After evaluating the model results, we note that there exist approaches that could be interesting to look at in a more comprehensive analysis. While this project focuses on exploring whether a relatively simple model using publicly available data can meaningfully reflect Vinmonopolet's store placement logic, future refinements could be valuable to understand decision making in a more nuanced way. For example, incorporating population projections rather than current population might offer more forward-looking insights, especially for long-term planning in municipalities with shifting demographics. Likewise, alternative demand proxies such as alcohol revenue rather than litres sold could potentially capture regional differences in consumption patterns that volume alone may overlook. While, for our purpose in this paper, these ideas lie outside the scope of our problem statement, they could be interesting directions to follow in future research on the field.

Summary

Overall, our model appears to perform reasonably well given the data constraints and scope of the task. The fact that Vinmonopolet has announced the opening of a new store in Aurland, ranked number four by probability in our model, despite our model omitting distance, offers tentative support for the model's practical relevance. Cross-validation confirms that the model generalizes relatively well, despite its simplicity and the limitations of the input data.

Our initial question asked if a model constructed using publicly available data could replicate or support Vinmonopolet's actual store establishment decisions. Based on the results, the approach is promising. Using a relatively simple model, we can generate predictions that align closely with known store openings. This supports the ideas discussed by Bresnahan and Reiss (1991). Even in data-limited environments, models based on broad, publicly available market characteristics can yield policy-relevant information regarding market entry decisions.

That said, there are limitations to be mentioned. Our model does not fully capture local non heterogeneity or institutional discretion such as unique geographic, cultural, or accessibility factors that might shape decisions in specific municipalities. Variables such as tourism and cross-border shopping were incorporated in simplified form and are likely noisy. Distance to existing stores, while conceptually important, was excluded to avoid contradictory inference. Thus, we argue that more granular or alternative data could improve model performance.

Still, as a complement to more qualitative or administrative assessments, we believe a data-driven model like ours could serve as a helpful reference, particularly when applied with a critical understanding of its assumptions. We are encouraged by the model performance despite its simplicity and acknowledge that future refinements such as incorporating better proxies for demand or local economic activity might improve the predictive accuracy further.

References

Aasestad. (2025, February 4). *12898: Overnattingar, etter region, innkvarteringstype, statistikkvariabel, år og bustadland. Statistikkbanken*. SSB. https://www.ssb.no/statbank/table/12898/

Berry, S. T. (1992). Estimation of a Model of Entry in the Airline Industry. *Econometrica*, *60*(4), 889. https://doi.org/10.2307/2951571

Boef, A. G. C., Dekkers, O. M., Vandenbroucke, J. P., & Le Cessie, S. (2014). Sample size importantly limits the usefulness of instrumental variable methods, depending on instrument strength and level of confounding. *Journal of Clinical Epidemiology*, *67*(11), 1258–1264. https://doi.org/10.1016/j.jclinepi.2014.05.019

Bresnahan, T. F., & Reiss, P. C. (1991). Entry and Competition in Concentrated Markets. *Journal of Political Economy*, *99*(5), 977–1009. https://doi.org/10.1086/261786

Bye, K., Grini, H., & Johnsen. (2025, February 26). *12852: Kommunefordelt månedslønn, etter region, statistikkmål, alder, avtalt/vanlig arbeidstid per uke, statistikkvariabel, år og arbeidssted/bosted. Statistikkbanken*. SSB. https://www.ssb.no/statbank/table/12852/

Einav, L., & Levin, J. (2010). Empirical Industrial Organization: A Progress Report. *The Journal of Economic Perspectives*, *24*(2), 145–162.

Haug, M. (2025, February 25). *07459: Befolkning, etter region, statistikkvariabel, år og alder. Statistikkbanken*. SSB. https://www.ssb.no/statbank/table/07459/

Henriksen, G., & Åsestad, K. (2025, February 11). *Grensehandel*. SSB. https://www.ssb.no/varehandel-og-tjenesteyting/varehandel/statistikk/grensehandel

Høydahl, E. (2025, February 26). *11342: Areal og befolkning, etter region, statistikkvariabel og år. Statistikkbanken*. SSB. https://www.ssb.no/statbank/table/11342/

Kartverket. (2024, November 28). *Stedsnavn WMS - Kartkatalogen*. https://kartkatalog.geonorge.no/metadata/stedsnavn-wms/c6eb0874-092a-4934-8ecb-d51da4c6f805?search=stedsnavn%20wms

Kartverket. (2025, April 4). *Administrative enheter kommuner—Kartkatalogen*. https://kartkatalog.geonorge.no/metadata/administrative-enheter-kommuner/041f1e6e-bdbc-4091-b48f-8a5990f3cc5b?search=administrative%20grenser%20kommuner

Kommuner uten pol. (2025, January 30). Retrieved 28 March 2025, from https://cdn.sanity.io/files/iv1p66d7/production/577b9bb9694f147a9251871741a9a4f4e64bc3c1.pdf

Lunheim, M. (2024, March 5). Dette sier vinmonopolet om å komme til Råde: Er på vei. *Moss-Avis*. https://www.moss-avis.no/dette-sier-vinmonopolet-om-a-komme-til-rade-er-pa-vei/s/5-67-1926334

Moe, D. (2022a, August 16). *Kommuneinndeling 2018*. https://www.ssb.no/klass/klassifikasjoner/131/versjon/833/endringer

Moe, D. (2022b, December 13). *Kommuneinndeling 2020*. https://www.ssb.no/klass/klassifikasjoner/131/versjon/1860/endringer

Moe, D. (2024a, July 12). *Kommuneinndeling 2017*. https://www.ssb.no/klass/klassifikasjoner/131/versjon/536/endringer

Moe, D. (2024b, October 1). *Kommuneinndeling 2024*. https://www.ssb.no/klass/klassifikasjoner/131/endringer

Nordahl, J. (2025, February 7). *Vinmonopolets salgstall 2024—Tabell butikkenes salg*. vinmonopolet.no. https://www.vinmonopolet.no/om-oss/salgstall/2024

Rapp, B.-L., & Bloch, V. V. H. (2024, October 1). *05212: Folkemengde, etter region, statistikkvariabel, år og tettbygd/spredtbygd. Statistikkbanken*. SSB. https://www.ssb.no/statbank/table/05212/

Vinmonopolet. (2025a, January 14). *Vårt samfunnssopdrag*. vinmonopolet.no. https://www.vinmonopolet.no/om-oss/vinmonopolordningen/samfunnsoppdrag

Vinmonopolet. (2025c, January 30). *Etablering av nye pol*. vinmonopolet.no. https://www.vinmonopolet.no/om-oss/drift/etablering-av-nye-vinmonopol

BAN440 - Term Paper Code

Candidate numbers: 74, 79, 85

Table of contents

Packages used

```
# relevant libraries
library(tidyverse)       # For data manipulation
library(readxl)          # For reading Excel files
library(fastDummies)     # For creating dummy variables
library(knitr)           # For creating tables
library(stargazer)       # For creating regression tables
library(caret)           # For data splitting and model evaluation
library(here)            # For file path management
library(httr)            # For API requests
library(jsonlite)        # For JSON parsing
library(readr)           # For reading CSV files
library(stringr)         # For string manipulation
library(tidyr)           # For unnesting
library(writexl)         # For writing Excel files
library(geosphere)       # For distance calculations
library(caret)           # For data splitting and model evaluation
library(kableExtra)      # For creating tables
library(tidymodels)      # For machine learning
library(sf)              # For spatial data handling
library(csmaps)          # For spatial data visualization
```

Data

Vinmonopolet API

Detailed descriptions of each Vinmonopolet store per 2024

```
# Define API URL
url <- "https://apis.vinmonopolet.no/stores/v0/details"

# Define your subscription key (replace with your actual key)
subscription_key <- "3b5b02c6793240fe9e6cb6d176e110e0"

# Send GET request with subscription key in header
response <- GET(url,
                add_headers(
```

```
                  Accept = "application/json",
                  `Ocp-Apim-Subscription-Key` = subscription_key  # API authentication
                ))


# Check response status
if (status_code(response) == 200) {
  # Convert API response to JSON and store it
  data <- content(response, as = "text", encoding = "UTF-8")
  store_data <- fromJSON(data)

  # View first few rows
  print(head(store_data))
} else {
  print(paste("Error:", status_code(response)))
}


# ---------------------------- Combine API with Vinmonopol Data ----------------------------


# Ensure store_data_clean is correctly formatted
store_data_clean <- store_data %>%
  unnest_wider(address) %>%  # Expands nested address fields
  select(
    storeId,
    storeName,
    status,
    postalCode,
    city,
    gpsCoord
  ) %>%
  rename(
    Store_ID = storeId,
    Store_Name = storeName,
    Store_Status = status,
    Postal_Code = postalCode,
    City = city,
    GPS_Coordinates = gpsCoord
  )
```

```r
# Transforming to normal characters
store_data_clean$Store_Name <- iconv(store_data_clean$Store_Name, from = "UTF-8", to = "ASC
store_data_clean$Store_Name <- trimws(store_data_clean$Store_Name)
```

Dimpostnummer merge

As stores have postal codes instead of municipality codes, we need a merge data set inbetween

```r
# Read the Dimpostnummer data
dimpostnummer_xlsx <- here("Data", "Vinmonopolet", "dimpostnummer.xlsx")

dimpostnummer_data <- read_excel(dimpostnummer_xlsx) %>%
  select("Postnummer", "Poststed", "Fylke", "KommuneKode", "Kommune")

# Merge with store_data_clean
store_data_clean <- store_data_clean %>%
  left_join(dimpostnummer_data, by = c("Postal_Code" = "Postnummer"))
```

Vinmonopolet 2024

The "foundation" with store names and sales data for 2024

```r
# Set locale to UTF-8
Sys.setlocale("LC_ALL", "en_US.UTF-8")

# Use here package to define the working directory
Vinmonopolet_2024 <- here("Data", "Vinmonopolet", "Vinmonopolet_2024.xlsx")

# Get the names of all sheets in the Excel file
sheet_names <- excel_sheets(Vinmonopolet_2024)

# Read each sheet into a list of data frames, skipping the first row
list_of_dfs <- lapply(sheet_names, function(sheet) {
  read_excel(Vinmonopolet_2024, sheet = sheet, skip = 2)
})

# Combine all data frames into a single data frame
```

```r
combined_data <- bind_rows(list_of_dfs)

# View the combined data frame
print(combined_data)

# Unique values in the first column
unique_values <- unique(combined_data$...1)

print(unique_values)

# Transforming to normal characters
combined_data$...1 <- iconv(combined_data$...1, from = "UTF-8", to = "ASCII//TRANSLIT")
combined_data$...1 <- trimws(combined_data$...1)

# Define the values to filter out
values_to_exclude <- c(
  "Svakvin", "Rodvin", "Hvitvin", "Musserende vin", "Rosevin",
  "Perlende vin", "Aromatisert vin", "Sider", "Fruktvin",
  "Brennevin", "Vodka", "Likor", "Whisky", "Akevitt",
  "Brennevin, annet", "Gin", "Druebrennevin",
  "Brennevin, noytralt < 37,5 %", "Rom", "Bitter",
  "Fruktbrennevin", "Genever", "Ol", "Alkoholfritt", "Sterkvin", "Totalsum",
  "eLager"
)

# Column names of combined data
colnames(combined_data)

# Filter out the specified values from the first column
filtered_data <- combined_data %>%
  mutate("2024" = as.numeric(`2024`),
         "Store" = as.character(`...1`)) %>%
  filter(!.[[1]] %in% values_to_exclude) %>%
  select("Store", "2024")

# Export the filtered data to an Excel file
#write_xlsx(filtered_data, "filtered_data.xlsx")

# Standardize store names to improve matching
filtered_data <- filtered_data %>%
```

```r
  mutate(Store = str_trim(str_to_lower(Store)))  # Trim spaces and convert to lowercase

store_data_clean <- store_data_clean %>%
  mutate(Store_Name = str_trim(str_to_lower(Store_Name)))  # Trim spaces and convert to low

# Remove unwanted characters from store names
store_data_clean <- store_data_clean %>%
  mutate(Store_Name = case_when(
    Store_Name == "oslo, thereses gate (stengt ja" ~ "oslo, thereses gate",
    Store_Name == "sandnes, sentrum" ~ "sandnes sentrum",
    Store_Name == "buvika" ~ "buvika, apent 24. oktober",
    Store_Name == "sola, tananger" ~ "sola, tananger, apnet 3. oktober",
    Store_Name == "oslo, bjorvika" ~ "oslo, bjorvika, apnet 14. mars 2024",
    Store_Name == "melhus" ~ "melhus, butikken stengt i 2023 pga kranvelt",
    Store_Name == "bergen, valkendorfsgt." ~ "bergen, valkendorfsgate",
    TRUE ~ Store_Name  # This keeps all other values unchanged
  ))

# Merge filtered_data (sales) with store_data_clean (store details)
final_data <- filtered_data %>%
  left_join(store_data_clean, by = c("Store" = "Store_Name"))  # Match by store name

# Check merged data
head(final_data)

# Write to excel
#write_xlsx(final_data, "final_data.xlsx")
```

Kommuneendringer 2017

```r
### Kommuneendringer 2017 ###

data_df <- final_data %>%
  rename(
    Municipality_Code = KommuneKode,
    Municipality_Name = Kommune
  )
```

```r
Kommuneendringer_17_xlsx <- here("Data", "Vinmonopolet", "Kommuneendringer_17.xlsx")

kommuneendringer_df <- read_excel(Kommuneendringer_17_xlsx)

# Clean up column names by using the correct column names
colnames(kommuneendringer_df) <- c("New_Codes", "Old_Codes")

# Split old municipality numbers into separate elements if they are separated by spaces
kommuneendringer_df$Old_Codes <- str_split(kommuneendringer_df$Old_Codes, " ")

# Extract the first four digits from each element in Old_Codes
old_codes_numeric <- lapply(kommuneendringer_df$Old_Codes, function(x) substr(x, 1, 4))

# Create a lookup list that maps old municipality codes to new codes
kommune_mapping <- setNames(rep(kommuneendringer_df$New_Codes, times = sapply(old_codes_num
                            unlist(old_codes_numeric))

# Update Municipality_Code and Municipality_Name in data_df
data_df <- data_df %>%
  rowwise() %>%
  mutate(
    new_val = if (Municipality_Code %in% names(kommune_mapping)) kommune_mapping[[Municipal
    Municipality_Code = if (!is.na(new_val)) substr(new_val, 1, 4) else Municipality_Code,
    Municipality_Name = if (!is.na(new_val)) {
      # Remove the municipality number and hyphen from the new value to get the municipalit
      str_trim(str_remove(new_val, "^[0-9]{4}\\s*-\\s*"))
    } else {
      Municipality_Name
    }
  ) %>%
  ungroup() %>%
  select(-new_val)

# Save the updated data to a new Excel file
#write_xlsx(data_df, "final_data_17.xlsx")
```

Kommuneendringer 2018

```
# Read in data from Excel files
Kommuneendringer_18_xlsx <- here("Data", "Vinmonopolet", "Kommuneendringer_18.xlsx")

kommuneendringer_df <- read_excel(Kommuneendringer_18_xlsx)

# Clean up column names by using the correct column names
colnames(kommuneendringer_df) <- c("New_Code", "Old_Codes")

# Split old municipality numbers (in case multiple old municipalities are separated by spac
kommuneendringer_df$Old_Codes <- str_split(kommuneendringer_df$Old_Codes, " ")

# Create a lookup list for old codes to new codes (one-way mapping)
kommune_mapping <- setNames(rep(kommuneendringer_df$New_Code, times = sapply(kommuneendring
                            unlist(kommuneendringer_df$Old_Codes))

# Update both Municipality_Code and Municipality_Name in data_df
data_df <- data_df %>%
  rowwise() %>%
  mutate(
    new_val = if (Municipality_Code %in% names(kommune_mapping)) kommune_mapping[[Municipal
    Municipality_Code = if (!is.na(new_val)) substr(new_val, 1, 4) else Municipality_Code,
    Municipality_Name = if (!is.na(new_val)) str_trim(str_remove(new_val, "^[0-9]{4}\\s*-\\
  ) %>%
  ungroup() %>%
  select(-new_val)

# Save the updated file
#write_xlsx(data_df, "final_data_18.xlsx")
```

Kommuneendringer 2020

```
# Read in data from Excel files
Kommuneendringer_20_xlsx <- here("Data", "Vinmonopolet", "Kommuneendringer_20.xlsx")

kommuneendringer_df <- read_excel(Kommuneendringer_20_xlsx)
```

```r
# Clean up column names by using the correct column names
colnames(kommuneendringer_df) <- c("New_Code", "Old_Codes")

# Split old municipality numbers (in case multiple old municipalities are separated by spac
kommuneendringer_df$Old_Codes <- str_split(kommuneendringer_df$Old_Codes, " ")

# Create a lookup list for old codes to new codes (one-way mapping)
kommune_mapping <- setNames(rep(kommuneendringer_df$New_Code, times = sapply(kommuneendring
                            unlist(kommuneendringer_df$Old_Codes))

# Update both Municipality_Code and Municipality_Name in data_df
data_df <- data_df %>%
  rowwise() %>%
  mutate(
    new_val = if (Municipality_Code %in% names(kommune_mapping)) kommune_mapping[[Municipal
    Municipality_Code = if (!is.na(new_val)) substr(new_val, 1, 4) else Municipality_Code,
    Municipality_Name = if (!is.na(new_val)) str_trim(str_remove(new_val, "^[0-9]{4}\\s*-\\
  ) %>%
  ungroup() %>%
  select(-new_val)

# Save the updated file
#write_xlsx(data_df, "final_data_20.xlsx")
```

Kommuneendringer 2024

```r
# Read in data from Excel files
Kommuneendringer_24_xlsx <- here("Data", "Vinmonopolet", "Kommuneendringer_24.xlsx")

kommuneendringer_df <- read_excel(Kommuneendringer_24_xlsx)


# Clean up column names by using the correct column names
colnames(kommuneendringer_df) <- c("New_Code", "Old_Codes")

# Split old municipality numbers (in case multiple old municipalities are separated by spac
kommuneendringer_df$Old_Codes <- str_split(kommuneendringer_df$Old_Codes, " ")
```

```r
# Create a lookup list for old codes to new codes (one-way mapping)
kommune_mapping <- setNames(rep(kommuneendringer_df$New_Code,
                                times = sapply(kommuneendringer_df$Old_Codes, length)),
                            unlist(kommuneendringer_df$Old_Codes))

# Update both Municipality_Code and Municipality_Name in data_df
data_df <- data_df %>%
  rowwise() %>%
  mutate(
    new_val = if (Municipality_Code %in% names(kommune_mapping)) kommune_mapping[[Municipal
    Municipality_Code = if (!is.na(new_val)) substr(new_val, 1, 4) else Municipality_Code,
    Municipality_Name = if (!is.na(new_val)) str_trim(str_remove(new_val, "^[0-9]{4}\\s*-\\
  ) %>%
  ungroup() %>%
  select(-new_val)

# Hardcode row 121 to set "Municipality_Code" to 1580 and "Municipality_Name" to Haram
data_df[121, "Municipality_Code"] <- "1580"
data_df[121, "Municipality_Name"] <- "Haram"

# Save the updated file
#write_xlsx(data_df, "final_data_24.xlsx")
```

Kommune 2025

Municipality data, including municipality number, population and Area

```r
# Kommune data file path
Kommune_data_xlsx <- here("Data", "Vinmonopolet", "Kommune_data.xlsx")

# Read data for total population and area of each municipality
kommune_data <- read_excel(Kommune_data_xlsx, skip = 3) %>%
  rename("Municipality" = "...1",
         "Population" = "2025...2",
         "Area" = "2025...3") %>%
  separate(Municipality, into = c("Mun_num", "Mun_name"), sep = " ", extra = "merge", fill
  filter(Population != 0,
         Area != 0) %>%
  mutate(Population = as.numeric(Population),
```

```
         Area = as.numeric(Area))

# Demographic data file path
Kommune_demo_xlsx <- here("Data", "Vinmonopolet", "Kommune_demo.xlsx")

# Read data for demographic data
demographic_data <- read_excel(Kommune_demo_xlsx, skip = 4) %>%
  rename("Municipality" = "...1",
         "0-17" = "0-17 år",
         "18+" = "18 år eller eldre") %>%
  filter(if_all(everything(), ~ !is.na(.) & . != 0)) %>% # Remove rows with NA or 0 in any
  separate(Municipality, into = c("Mun_num", "Mun_name"), sep = " ", extra = "merge", fill
  separate(Mun_num, into = c("K", "Mun_num"), sep = "-") %>%
  select(-"K",
         -"Mun_name")

# Merge the two datasets
kommune_data_final <- kommune_data %>%
  left_join(demographic_data, by = c("Mun_num"))

# Write data to Excel
#write_xlsx(kommune_data_final, "Kommune_data_final.xlsx")
```

Demography data

```
final_data <- data_df

# Transforming to normal characters
final_data$Municipality_Name <- iconv(final_data$Municipality_Name, from = "UTF-8", to = "A

final_data$Municipality_Name <- trimws(final_data$Municipality_Name)

# Standardize store names to improve matching
final_data <- final_data %>%
  mutate(Municipality_Name = str_trim(str_to_lower(Municipality_Name)))  # Trim spaces and

# Loading the kommune data
kommune_data <- kommune_data_final
```

```r
# Standardize the kommune data
kommune_data <- kommune_data %>%
  mutate(Mun_name = iconv(Mun_name, from = "UTF-8", to = "ASCII//TRANSLIT"),
         Mun_name = str_trim(str_to_lower(Mun_name)))  # Trim spaces and convert to lowerca

# Perform a full join to include all rows from both datasets
merged_data <- final_data %>%
  full_join(kommune_data, by = c("Municipality_Code" = "Mun_num"))

# Replace NA values in store-related columns with 0
# Assuming 'Store_Info_Column' is the column in final_data that contains store information
# Replace 'Store_Info_Column' with the actual column names you want to fill with 0
merged_data <- merged_data %>%
  mutate(across(where(is.numeric), ~ replace_na(.x, 0)))

# If you have specific columns to replace NA with 0, you can specify them like this:
# merged_data <- merged_data %>%
#   mutate(Store_Info_Column = replace_na(Store_Info_Column, 0))

# Write the merged data to an Excel file
#write_xlsx(merged_data, "final_data_mun.xlsx")
```

Distance data

This is just our code for the calculation of dist_nearest. As the actual data file is too large to submit, we jump to the next step with the resulting data saved as "final_data_mun_dist.xlsx"

```r
# ----------------------------
# 1. Load and Prepare Data
# ----------------------------
# Load Vinmonopolet + municipality dataset
#data <- read_excel("final_data_mun.xlsx")

# Load pre-cleaned municipality admin center coordinates
#admin_centers_final <- readRDS("admin_centers_final.rds")

# Ensure join columns match in type
#data <- data %>%
```

```
#  mutate(Municipality_Code = as.character(Municipality_Code))

#admin_centers_final <- admin_centers_final %>%
#  mutate(kommunenummer = as.character(kommunenummer))

# ------------------------------
# 2. Merge Coordinates
# ------------------------------
# Merge admin center lat/lon into dataset by municipality
#data <- left_join(data, admin_centers_final, by = c("Municipality_Code" = "kommunenummer")

# Overwrite old coordinates with admin center coordinates
#data <- data %>%
#  mutate(
#    Latitude = as.numeric(lat),
#    Longitude = as.numeric(lon)
#  )

# ------------------------------
# 3. Parse Store GPS Coordinates
# ------------------------------
# Split store GPS into separate numeric lat/lon

# ---------------------------
# STEP 1: Load dataset
# ---------------------------

# Read merged dataset with both Vinmonopolet store info and municipality info
#data <- read_excel("final_data_mun.xlsx")

# ---------------------------
# STEP 2: Parse store coordinates
# ---------------------------

# GPS_Coordinates column contains both latitude and longitude as a string separated by ";"
# We split this into two separate numeric columns: store_lat and store_lon

#data <- data %>%
#  separate(GPS_Coordinates, into = c("store_lat", "store_lon"), sep = ";", convert = TRUE)
#  mutate(
```

```
#    store_lat = as.numeric(store_lat),   # ensure store latitude is numeric
#    store_lon = as.numeric(store_lon)    # ensure store longitude is numeric
#  )


# -----------------------------
# 4. Build Store Location Matrix
# -----------------------------
# Extract distinct (lon, lat) of all Vinmonopolet stores
#store_locations <- data %>%
#  filter(!is.na(store_lon), !is.na(store_lat)) %>%

# ---------------------------
# STEP 3: Ensure municipality center coordinates are numeric
# ---------------------------

# These are already separate in the dataset, but stored as characters - we convert them
#data <- data %>%
#  mutate(
#    Longitude = as.numeric(Longitude),  # longitude of the municipality center
#    Latitude = as.numeric(Latitude)     # latitude of the municipality center
#  )

# ---------------------------
# STEP 4: Extract store coordinates for distance calculation
# ---------------------------

# We only want to use valid store locations for calculating distances
# (some rows in the dataset are just municipality data with no store info)
#store_data <- data %>%
#  filter(!is.na(store_lat), !is.na(store_lon))

# Extract a unique matrix of all Vinmonopolet store locations
# Format required by geosphere is matrix of (longitude, latitude)
#store_locations <- store_data %>%

#  select(store_lon, store_lat) %>%
#  distinct() %>%
#  as.matrix()
```

```
# ---------------------------
# STEP 5: Define function to calculate distance to nearest store
# ---------------------------

# For a given municipality center (lon, lat), compute distance to nearest store
# Uses Haversine formula (accounts for Earth's curvature)
#min_distance_to_store <- function(lon, lat) {
#  if (is.na(lon) || is.na(lat)) {
#    return(NA)  # return NA if municipality coordinates are missing
#  }
#  muni_coord <- matrix(c(lon, lat), nrow = 1)  # convert to matrix format for geosphere
#  dists <- distHaversine(muni_coord, store_locations)  # distances in meters
#  return(min(dists) / 1000)  # convert to kilometers
#}

# ---------------------------
# STEP 6: Apply distance function to each municipality
# ---------------------------

# For each row (i.e., each municipality center), calculate distance to closest Vinmonopolet
# Note: This includes all rows (even ones without a store)

#data$dist_nearest_store <- mapply(
#  min_distance_to_store,
#  data$Longitude,
#  data$Latitude
#)



# ---------------------------
# STEP 7: Quick check (optional)
# ---------------------------

# Check that coordinates are numeric
#str(data$Longitude)
#str(data$Latitude)



# -----------------------------
# 7. Optional: Drop Redundant Columns
```

```
# ------------------------------
#data <- data %>%
#  select(
#    -lat, -lon, -multikurve, -kommunenavn
#  )

# ------------------------------
# 8. Final Checks (Optional)
# ------------------------------
#str(data$dist_nearest_store)
#summary(data$dist_nearest_store)

# ------------------------------
# 9. Does VInmonopolets 30 km threshold 97% goal work based on our data
# ------------------------------
# 1. Total population (all municipalities)
#total_pop <- sum(data$Population, na.rm = TRUE)

# 2. Population in municipalities with distance > 30 km
#pop_far_away <- data %>%
#  filter(dist_nearest_store > 30) %>%
#  summarise(total = sum(Population, na.rm = TRUE)) %>%
#  pull(total)

# 3. Share of population far away
#share_far_away <- pop_far_away / total_pop

# 4. Share WITH access (within 30 km)
#share_within_30km <- 1 - share_far_away

# 5. Print results
#cat(sprintf("Share of population within 30 km of a Vinmonopolet: %.2f%%\n", #share_within_
#cat(sprintf("Target (Vinmonopolet): 97%%\n"))

#underserved <- data %>%
#  filter(dist_nearest_store > 30) %>%
#  select(,Mun_name, Population, dist_nearest_store) %>%
#  arrange(desc(dist_nearest_store))

#print(underserved, n = 50)
```

```
# ----------------------------
# 10. Export the final data to an Excel file
# ----------------------------

#library(writexl)
#write_xlsx(data, "final_data_mun_dist.xlsx")
# ----------------------------
```

Model variables merge

```
### Independent variables merge ###

final_data_mun_dist <- here("Data", "Vinmonopolet", "final_data_mun_dist.xlsx")

# Load data
Vinmonopolet <- read_excel(final_data_mun_dist) %>%
  select(-c(Store_ID, Store_Status, Postal_Code, Poststed,
            PostnummerKategoriKode, PostnummerKategori, Region_Code,
            Municipality_Name)) %>%
  mutate(
    Municipality_Name = Mun_name,
    Region_Name = case_when(
      Region_Name == "AUST-AGDER" ~ "Agder",
      Region_Name == "VEST-AGDER" ~ "Agder",
      Region_Name == "AKERSHUS" ~ "Akershus",
      Region_Name == "OPPLAND" ~ "Innlandet",
      Region_Name == "BUSKERUD" ~ "Buskerud",
      Region_Name == "VESTFOLD" ~ "Vestfold",
      Region_Name == "FINNMARK" ~ "Finnmark",
      Region_Name == "HEDMARK" ~ "Innlandet",
      Region_Name == "MØRE OG ROMSDAL" ~ "Møre og Romsdal",
      Region_Name == "NORDLAND" ~ "Nordland",
      Region_Name == "OSLO" ~ "Oslo",
      Region_Name == "ROGALAND" ~ "Rogaland",
      Region_Name == "TELEMARK" ~ "Telemark",
      Region_Name == "TROMS" ~ "Troms",
      Region_Name == "SØR-TRØNDELAG" ~ "Trøndelag",
      Region_Name == "NORD-TRØNDELAG" ~ "Trøndelag",
```

```
      Region_Name == "SOGN OG FJORDANE" ~ "Vestland",
      Region_Name == "HORDALAND" ~ "Vestland",
      Region_Name == "ØSTFOLD" ~ "Østfold",
      is.na(Region_Name) & str_starts(Municipality_Code, "03") ~ "Oslo",
      is.na(Region_Name) & str_starts(Municipality_Code, "11") ~ "Rogaland",
      is.na(Region_Name) & str_starts(Municipality_Code, "15") ~ "Møre og Romsdal",
      is.na(Region_Name) & str_starts(Municipality_Code, "18") ~ "Nordland",
      is.na(Region_Name) & str_starts(Municipality_Code, "31") ~ "Østfold",
      is.na(Region_Name) & str_starts(Municipality_Code, "32") ~ "Akershus",
      is.na(Region_Name) & str_starts(Municipality_Code, "33") ~ "Buskerud",
      is.na(Region_Name) & str_starts(Municipality_Code, "34") ~ "Innlandet",
      is.na(Region_Name) & str_starts(Municipality_Code, "39") ~ "Vestfold",
      is.na(Region_Name) & str_starts(Municipality_Code, "40") ~ "Telemark",
      is.na(Region_Name) & str_starts(Municipality_Code, "42") ~ "Agder",
      is.na(Region_Name) & str_starts(Municipality_Code, "46") ~ "Vestland",
      is.na(Region_Name) & str_starts(Municipality_Code, "50") ~ "Trøndelag",
      is.na(Region_Name) & str_starts(Municipality_Code, "55") ~ "Troms",
      is.na(Region_Name) & str_starts(Municipality_Code, "56") ~ "Finnmark",
      TRUE ~ Region_Name  # Keep existing Region_Name if no conditions are met
    )
  ) %>%
  select(-Mun_name)

# Aggregating per municipality data
Vinmonopolet_market <- Vinmonopolet %>%
  group_by(Municipality_Code) %>%
  summarise(
    Mun_name = first(Municipality_Name),
    Region_Name = first(Region_Name),
    Population = first(Population),
    Area = first(Area),
    Number_of_stores = sum(`2024` > 0),  # Count non-zero sales
    Sales = sum(`2024`),
    Lat = first(Latitude),
    Lon = first(Longitude),
    Dist_nearest = first(dist_nearest_store),
  )


# Scaling the variables that have nt been scaled yet
```

```r
Vinmonopolet_market <- Vinmonopolet_market %>%
  mutate(Population = Population / 1000,
         Sales = Sales / 1000)

# Now we have loaded and wrangled the main data set, but we can use some
# new variables for our analysis

## Merge 1: Grensehandel ##########################################################

Grensehandel_weights <- here("Data", "Vinmonopolet", "Grensehandel_weights.xlsx")

# Load the weights datas
weights <- read_excel(Grensehandel_weights, skip = 3) %>%
  slice(1) %>%
  select(-'...1') %>%
  mutate(
    mean_weight = (as.numeric(`2024K1`) + as.numeric(`2024K2`) + as.numeric(`2024K3`) + as.
  )

weight_grensehandel <- weights$mean_weight / 100

# Load the regional data
Grensehandel_regions <- here("Data", "Vinmonopolet", "Grensehandel_regions.xlsx")

regional <- read_excel(Grensehandel_regions)

total_grensehandel <- sum(regional$"2024")

# Calculate grensehandel per region
regional <- regional %>%
  rename(
    Region = `Fylker`,
    Total_sale = `2024`
  ) %>%
  mutate(
    Grensehandel = Total_sale * weight_grensehandel
  )

# Split the "Vestlandet" region row into three new rows: "Rogaland", "Vestland" and "MC8re
regional <- regional %>%
```

```
  rbind(
    regional %>% filter(Region == "Vestlandet") %>% mutate(Region = "Rogaland"),
    regional %>% filter(Region == "Vestlandet") %>% mutate(Region = "Vestland"),
    regional %>% filter(Region == "Vestlandet") %>% mutate(Region = "Møre og Romsdal")
  ) %>%
  filter(Region != "Vestlandet")

# Divide the grensehandel value by three for "Rogaland", "Vestland" and "MC8re og Romsdal"
regional <- regional %>%
  mutate(
    Grensehandel = case_when(
      Region == "Rogaland" ~ Grensehandel * 0.35,
      Region == "Vestland" ~ Grensehandel * 0.46,
      Region == "Møre og Romsdal" ~ Grensehandel * 0.19,
      TRUE ~ Grensehandel  # Keep the original value for other regions
    )
  )

# Split the "Nord-Norge" region row into three new rows: "Nordland", "Troms" and "Finnmark"
# And divide the grensehandel value by three
regional <- regional %>%
  mutate(
    Grensehandel = ifelse(Region == "Nord-Norge", Grensehandel / 3, Grensehandel)
  ) %>%
  rbind(
    regional %>% filter(Region == "Nord-Norge") %>% mutate(Region = "Nordland"),
    regional %>% filter(Region == "Nord-Norge") %>% mutate(Region = "Troms"),
    regional %>% filter(Region == "Nord-Norge") %>% mutate(Region = "Finnmark")
  ) %>%
  filter(Region != "Nord-Norge")

# Divide the grensehandel value by three for "Nordland", "Troms" and "Finnmark"
regional <- regional %>%
  mutate(
    Grensehandel = case_when(
      Region == "Nordland" ~ Grensehandel * 0.5,
      Region == "Troms" ~ Grensehandel * 0.35,
      Region == "Finnmark" ~ Grensehandel * 0.15,
      TRUE ~ Grensehandel  # Keep the original value for other regions
    )
```

```
  )

# Split the "Agder, Telemark, Buskerud og Vestfold" column into four new columns: "Agder",
# And divide the grensehandel value by four
regional <- regional %>%
  mutate(
    Grensehandel = ifelse(Region == "Agder, Telemark, Buskerud og Vestfold", Grensehandel /
  ) %>%
  rbind(
    regional %>% filter(Region == "Agder, Telemark, Buskerud og Vestfold") %>% mutate(Regio
    regional %>% filter(Region == "Agder, Telemark, Buskerud og Vestfold") %>% mutate(Regio
    regional %>% filter(Region == "Agder, Telemark, Buskerud og Vestfold") %>% mutate(Regio
    regional %>% filter(Region == "Agder, Telemark, Buskerud og Vestfold") %>% mutate(Regio
  ) %>%
  filter(Region != "Agder, Telemark, Buskerud og Vestfold")

# Divide the grensehandel value by four for "Agder", "Telemark", "Buskerud" and "Vestfold"
regional <- regional %>%
  mutate(
    Grensehandel = case_when(
      Region == "Agder" ~ Grensehandel * 0.31,
      Region == "Telemark" ~ Grensehandel * 0.17,
      Region == "Buskerud" ~ Grensehandel * 0.26,
      Region == "Vestfold" ~ Grensehandel * 0.26,
      TRUE ~ Grensehandel  # Keep the original value for other regions
    )
  )

# Removing the "total_sale" column from the regional data set
regional <- regional %>% select(-Total_sale)

# Merge the regional data with the main data set on Region_Name in the Vinmonopolet_market
Vinmonopolet_market <- left_join(Vinmonopolet_market, regional, by = c("Region_Name" = "Reg

# Add a new column "Region_pop" where "Population" is summarized for each region
Vinmonopolet_market <- Vinmonopolet_market %>%
  group_by(Region_Name) %>%
  mutate(Region_pop = sum(Population)) %>%
  ungroup()
```

```r
Vinmonopolet_market <- Vinmonopolet_market %>%
  mutate(Kommune_share = Population / Region_pop,
         Grensehandel_mun = Grensehandel * Kommune_share) %>%
  select(-c("Region_pop", "Kommune_share", "Grensehandel")) %>%
  rename(Grensehandel = Grensehandel_mun)




## Merge 2: Tourism ################################################################

Tourism_xlsx <- here("Data", "Vinmonopolet", "Tourism.xlsx")

# Reading tourism data
Tourism <- read_excel(Tourism_xlsx, skip = 4) %>%
  rename(
    Mun = '...1',
    H = 'Hotell og liknande overnattingsbedrifter',
    C = 'Campingplassar, hyttegrender og vandrarheim',
  ) %>%
  select(-'...2') %>%
  mutate_at(vars(H, C), ~as.numeric(str_replace_all(., ":", "0"))) %>%
  mutate(n_stays = H + C) %>%
  separate(Mun, into = c("Municipality_Code", "Municipality_Name"), sep = " ", remove = FAL
  select(-c("Mun", "H", "C", "Municipality_Name")) %>%
  filter(!is.na(Municipality_Code))

# Merging the data
Vinmonopolet_market <- left_join(Vinmonopolet_market, Tourism, by = "Municipality_Code") %>
  mutate(
    n_stays = ifelse(is.na(n_stays), 0, n_stays),
    n_stays = n_stays / 1000
  )

# There is a great deal of missing data, so we do not know the relevance of
# this data yet


## Merge 3: Income #################################################################
```

```r
# Average monthly salary per inhabitant in the municipality

# Load data
Monthly_Salary <- here("Data", "Vinmonopolet", "Monthly_Salary.xlsx")

data <- read_excel(Monthly_Salary)

# Cleaning data by removing rows with missing values and rows with dots
clean_data <- data %>%
  filter(!apply(., 1, function(row) any(grepl("\\.", row)))) %>%
  na.omit()

# Remove the last two rows from the data, using tidyverse
clean_data <- clean_data %>%
  slice(1:(n() - 2)) %>%
  select(-'...2') %>%
  rename(
    Mun = `12852: Kommunefordelt månedslønn, etter region, statistikkmål, statistikkvariabe
    Monthly_salary = '...3'
  ) %>%
  separate(Mun, into = c("Municipality_Code", "Municipality_Name"), sep = " ", remove = FAL
  select(-c("Municipality_Name", "Mun")) %>%
  mutate(Monthly_salary = as.numeric(Monthly_salary),
         Monthly_salary = Monthly_salary / 1000)

# Merge with the main data set
Vinmonopolet_market <- left_join(Vinmonopolet_market, clean_data, by = "Municipality_Code")




## Merge 4: Concentration ##########################################################

# Load data
Concentration_xlsx <- here("Data", "Vinmonopolet", "Concentration.xlsx")

concentration <- read_excel(Concentration_xlsx, skip = 5) %>%
  slice(1:357) %>%
  select('...1',
```

```
         'Spredtbygd strøk...3') %>%
  rename(Mun = '...1',
         Spread = 'Spredtbygd strøk...3') %>%
  separate(Mun, into = c("Municipality_Code", "Municipality_Name"), sep = " ", remove = FAL
  select(-c("Municipality_Name", "Mun")) %>%
  mutate(Spread = as.numeric(Spread),
         Spread = Spread / 1000)

# Remove the first two characters of each cell in the "Municpality_Code" column
concentration$Municipality_Code <- substr(concentration$Municipality_Code, 3, nchar(concent

# Merge with the main data set
Vinmonopolet_market <- left_join(Vinmonopolet_market, concentration, by = "Municipality_Cod


## Merge 5: "Active" stores ##########################################################

# Load data
Active_xlsx <- here("Data", "Vinmonopolet", "Active.xlsx")

A1 <- read_excel(Active_xlsx, sheet = 1, skip = 2)

A2 <- read_excel(Active_xlsx, sheet = 2, skip = 2)

# Merge the two data sets
Active <- A1 %>%
  bind_rows(A2) %>%
  select(-c('1', '...3', Fylke))

# Rename columns
names(Active)[1] <- "Mun_name"

# Remove unncessary spaces and numbers from the "Mun_name" column
Active$Mun_name <- substr(Active$Mun_name, 4, nchar(Active$Mun_name))

Active$Mun_name <- trimws(Active$Mun_name, which = "left")

# Replace norwegian special letters with english ones and make all letters lowercase
Active$Mun_name <- tolower(iconv(Active$Mun_name, from = "UTF-8", to = "ASCII//TRANSLIT"))
```

```r
# Recode the "Mun_name" column
Active$Mun_name <- case_when(
  Active$Mun_name == "hamaroy" ~ "habmer - hamaroy",
  Active$Mun_name == "hattfjelldal" ~ "aarborte - hattfjelldal",
  Active$Mun_name == "valer (viken)" ~ "valer (ostfold)",
  TRUE ~ Active$Mun_name)

# Merge with the main data set

# Make a dummy variable for active stores
Vinmonopolet_market$Active <- ifelse(Vinmonopolet_market$Mun_name %in% Active$Mun_name, 1,

## Write to Excel ##################################################################

# Write to Excel
# write_xlsx(Vinmonopolet_market, "demand_data.xlsx")
```

Model applications

Data preparation

```r
### Data preparation ###############################################################

# Rename relevant columns in accordance with tidyverse standards
Vinmonopolet_market <- Vinmonopolet_market %>%
  rename(
    mun_code = Municipality_Code,
    mun_name = Mun_name,
    region_name = Region_Name,
    population = Population,
    area = Area,
    number_of_stores = Number_of_stores,
    sales = Sales,
    lat = Lat,
    lon = Lon,
    dist_nearest = Dist_nearest,
    grensehandel = Grensehandel,
    n_stays = n_stays,
```

```
    monthly_salary = Monthly_salary,
    spread = Spread,
    active = Active
  )


# Narrowing down the data to only contain relevant markets
# Excluding the largest cities because they are not representative

# Train and test split, training data all observations with a store
train_data <- Vinmonopolet_market %>%
  filter(number_of_stores > 0)

# Test data all observations without a store
test_data <- Vinmonopolet_market %>%
  filter(number_of_stores == 0)
```

Model selection and basic regressions

```
### Model selection ################################################################

# Forward selection
forward_model <- step(lm(sales ~ 1, data = train_data),
                      scope = ~ population + grensehandel + n_stays + monthly_salary + area
                      direction = "forward")

#summary(forward_model)

# Backward selection
backward_model <- step(lm(sales ~ population + grensehandel + n_stays + monthly_salary + ar
                          data = train_data),
                       direction = "backward")

#summary(backward_model)


lm_Area <- lm(sales ~ area, data = train_data)
```

```r
#summary(lm_Area)

lm_pop <- lm(sales ~ population, data = Vinmonopolet_market)

#summary(lm_pop)

small_data <- Vinmonopolet_market %>%
  filter(number_of_stores == 1 | 0)

lm_pop_test <- lm(sales ~ population, data = small_data)

#summary(lm_pop_test)

# Linear regression model for predicting sales with all the variables
var_test <- lm(sales ~ population + grensehandel + n_stays + monthly_salary + area +
                 number_of_stores + spread,
               data = Vinmonopolet_market)

#stargazer(var_test, type = "text")

# From these regressions we see that we want to remove the "Area" and "prop_spread" variabl
# from the regressions as they are not significant.
```

Demand estimation

```
### Demand estimation ##################################################################

## Linear regression

# Predicting sales using the training data
reg1 <- lm(sales ~ population + grensehandel + n_stays + monthly_salary,
           data = train_data)

stargazer(reg1, type = "text")
```

```
================================================
                        Dependent variable:
                    ----------------------------
                               sales
------------------------------------------------
population                   16.449***
                              (0.490)

grensehandel                 -5.264***
                              (1.456)

n_stays                      0.246***
                              (0.043)

monthly_salary                4.885**
                              (2.290)

Constant                    -270.085**
                             (125.336)

------------------------------------------------
Observations                    237
R2                             0.990
Adjusted R2                    0.990
Residual Std. Error     102.638 (df = 232)
F Statistic         5,683.347*** (df = 4; 232)
```

```
================================================
Note:                *p<0.1; **p<0.05; ***p<0.01
```

```r
# Applying the model on the test data
test_data$sales_pred <- predict(reg1, newdata = test_data)

## Merge predicted data into the original data

# Deselect unnecessary columns to merge the data easier
test_data <- test_data %>%
  select(mun_code, sales_pred)

# Merge predicted demand (sales) back into the original data
Vinmonopolet_market <- Vinmonopolet_market %>%
  left_join(test_data, by = "mun_code") %>%
  mutate(sales = ifelse(sales == 0, sales_pred, sales)) %>%
  select(-sales_pred) %>%
  mutate(sales = ifelse(sales < 0, 0, sales),
         number_of_stores = as.integer(number_of_stores)) %>%
  filter(number_of_stores < 2)
```

Logit model

```
## Logit regression ##############################################################

# Make sure the factor for Number_of_stores has valid R variable names
# that won't cause errors in caret. For instance, rename "0" -> "NoStore"
# and "1" -> "OneStore".
data_for_logit <- Vinmonopolet_market %>%
  mutate(number_of_stores = as.factor(number_of_stores))

# Rename factor levels (originally "0" and "1") to "NoStore" and "OneStore"
data_for_logit$number_of_stores <- factor(
  data_for_logit$number_of_stores,
  levels = c("0", "1"),
  labels = c("NoStore", "OneStore")
)

# Set up k-fold cross-validation parameters
set.seed(123)  # for reproducibility

my_control <- trainControl(
  method = "cv",            # k-fold CV
  number = 5,               # 5 folds
  classProbs = TRUE,        # needed for probability output
  summaryFunction = twoClassSummary
)

# Train the logistic model with cross-validation
cv_model <- train(
  number_of_stores ~ sales,
  data = data_for_logit,
  method = "glm",
  family = binomial,
  trControl = my_control,
  metric = "ROC"            # use AUC (Area Under the Curve) as our metric
)

# Review cross-validation results
print(cv_model)
```

```
Generalized Linear Model

316 samples
  1 predictor
  2 classes: 'NoStore', 'OneStore'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 252, 253, 253, 253, 253
Resampling results:

  ROC        Sens       Spec
  0.9397703  0.8666667  0.8670513
```

```
print(cv_model$results)
```

```
  parameter       ROC      Sens      Spec      ROCSD    SensSD     SpecSD
1      none 0.9397703 0.8666667 0.8670513 0.02518379 0.1078515 0.04987575
```

```
# Get predicted probabilities from the final trained model
# caret retrains on the entire dataset after CV by default
Vinmonopolet_market$prob <- predict(cv_model, newdata = data_for_logit, type = "prob")[, "O

# For recommended stores
recommended_stores <- Vinmonopolet_market %>%
  mutate(number_of_stores = as.integer(as.character(number_of_stores))) %>%
  filter(number_of_stores == 0, dist_nearest > 0) %>%
  arrange(desc(prob)) %>%
  select(mun_name, prob, dist_nearest, sales, population, region_name, active) %>%
  mutate(
    dist_nearest = round(dist_nearest, 0),  # Round dist_nearest to 0 decimals
    sales = round(sales, 0),                # Round sales to 0 decimals
    across(where(is.numeric) & !c(dist_nearest, sales), ~round(., 3))  # Round other numeri
  )

# Create table
kable(head(recommended_stores, 10),
      format = "markdown",
      digits = 3,
```

```
    longtable = TRUE) %>%
kable_styling(latex_options = "scale_down")
```

mun_name	prob	dist_nearest	sales	population	region_name	active
giske	0.995	3	156	8.773	Møre og Romsdal	0
lunner	0.993	7	150	9.420	Akershus	1
rade	0.967	13	118	7.850	Østfold	0
hareid	0.857	10	87	5.320	Møre og Romsdal	1
valer (ostfold)	0.823	11	82	6.162	Østfold	1
aurland	0.817	37	81	1.836	Vestland	1
birkenes	0.752	13	74	5.413	Agder	0
eidskog	0.729	23	71	6.059	Innlandet	1
aure	0.683	28	67	3.394	Møre og Romsdal	1
austrheim	0.658	16	65	2.915	Vestland	1

```r
# For active stores
Active_stores <- Vinmonopolet_market %>%
  mutate(number_of_stores = as.integer(as.character(number_of_stores))) %>%
  filter(active == 1, dist_nearest > 0) %>%
  arrange(desc(prob)) %>%
  select(mun_name, prob, dist_nearest, sales, population, region_name, active) %>%
  mutate(
    dist_nearest = round(dist_nearest, 0),  # Round dist_nearest to 0 decimals
    sales = round(sales, 0),                # Round sales to 0 decimals
    across(where(is.numeric) & !c(dist_nearest, sales), ~round(., 3))  # Round other numeri
  )

# Create table
kable(head(Active_stores, 10),
      format = "markdown",
      digits = 3,
      longtable = TRUE) %>%
  kable_styling(latex_options = "scale_down")
```

mun_name	prob	dist_nearest	sales	population	region_name	active
lunner	0.993	7	150	9.420	Akershus	1
hareid	0.857	10	87	5.320	Møre og Romsdal	1
valer (ostfold)	0.823	11	82	6.162	Østfold	1
aurland	0.817	37	81	1.836	Vestland	1
eidskog	0.729	23	71	6.059	Innlandet	1
aure	0.683	28	67	3.394	Møre og Romsdal	1
austrheim	0.658	16	65	2.915	Vestland	1
aukra	0.644	15	64	3.759	Møre og Romsdal	1
vaksdal	0.617	19	61	3.875	Vestland	1
sokndal	0.544	21	55	3.371	Rogaland	1

```r
recommended_stores_dist <- Vinmonopolet_market %>%
  mutate(number_of_stores = as.integer(as.character(number_of_stores))) %>%
  filter(number_of_stores == 0, dist_nearest > 0) %>%
  arrange(desc(dist_nearest)) %>%   # Sort by dist_nearest, descending
  select(mun_name, prob, dist_nearest, sales, population, region_name, active) %>%
  mutate(
    dist_nearest = round(dist_nearest, 0),  # Round dist_nearest to 0 decimals
    sales = round(sales, 0),                # Round sales to 0 decimals
    across(where(is.numeric) & !c(dist_nearest, sales), ~round(., 3))  # Round other numeri
  )

# Create table
kable(head(recommended_stores_dist, 10),
      format = "markdown",
      digits = 3,
      longtable = TRUE) %>%
  kable_styling(latex_options = "scale_down")
```

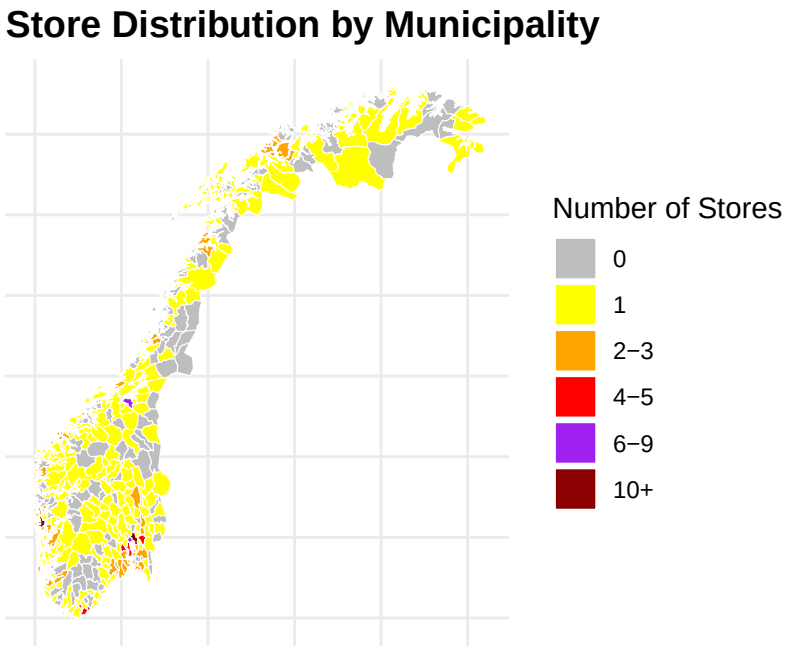
mun_name	prob	dist_nearest	sales	population	region_name	active
rost	0.066	94	0	0.458	Nordland	1
raarvihke - royrvik	0.066	76	0	0.443	Trøndelag	0
karasjohka - karasjok	0.243	67	30	2.524	Finnmark	1
varoy	0.066	66	0	0.677	Nordland	0
namsskogan	0.066	65	0	0.811	Trøndelag	1
lierne	0.084	61	5	1.316	Trøndelag	0
deatnu - tana	0.280	61	33	2.798	Finnmark	0
trana	0.066	53	0	0.463	Nordland	1
engerdal	0.066	52	0	1.326	Innlandet	1
hasvik	0.066	52	0	0.977	Finnmark	1

Visualisations

```r
# Use the existing municipality map data
municipalities <- nor_municip_map_b2024_default_sf

# Read your store dataset
```

```r
store_data_path <- here("Data", "Vinmonopolet", "final_data_24.xlsx")
store_data <- readxl::read_excel(store_data_path)

# Extract the last 4 digits from "location_code" to get the municipality code
municipalities <- municipalities %>%
  mutate(municip_code = as.integer(str_sub(location_code, -4)))  # Extract last 4 digits

# Count the number of stores per municipality
store_counts <- store_data %>%
  group_by(Municipality_Code) %>%
  summarise(num_stores = n())

# Changing to matching format
store_counts$Municipality_Code <- as.numeric(store_counts$Municipality_Code)

# Merge store counts with the geographical dataset
merged_data <- municipalities %>%
  left_join(store_counts, by = c("municip_code" = "Municipality_Code"))

# Assign store categories, treating NA as 0
merged_data <- merged_data %>%
  mutate(store_category = case_when(
    is.na(num_stores) ~ "0",
    num_stores == 1 ~ "1",
    num_stores %in% 2:3 ~ "2-3",
    num_stores %in% 4:5 ~ "4-5",
    num_stores %in% 6:9 ~ "6-9",
    num_stores >= 10 ~ "10+",
    TRUE ~ NA_character_
  ))

# Make sure factor levels are ordered correctly
merged_data$store_category <- factor(
  merged_data$store_category,
  levels = c("0", "1", "2-3", "4-5", "6-9", "10+")
)

# Plot the heatmap with discrete color bins
ggplot(merged_data) +
  geom_sf(aes(fill = store_category), color = "white", size = 0.1) +
```

```
  scale_fill_manual(
    values = c("0" = "grey", "1" = "yellow", "2-3" = "orange", "4-5" = "red", "6-9" = "purp
    name = "Number of Stores"
  ) +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank()) +
  labs(title = "Store Distribution by Municipality",
       x = NULL, y = NULL)
```




```
# Extract the last 4 digits from "location_code" to get the municipality code
municipalities <- municipalities %>%
  mutate(municip_code = as.integer(str_sub(location_code, -4)))  # Extract last 4 digits

# Merge population data with the existing geographical dataset
Vinmonopolet_market$mun_code <- as.numeric(Vinmonopolet_market$mun_code)
```

```r
# Add a "prob_category" for municipalities that have at least one store
Vinmonopolet_market <- Vinmonopolet_market %>%
  mutate(prob_category = case_when(
    number_of_stores > 0 ~ "Has a store",
    prob >= 0 & prob < 0.25 ~ "Low",
    prob >= 0.25 & prob < 0.5 ~ "Medium Low",
    prob >= 0.5 & prob < 0.75 ~ "Medium High",
    prob >= 0.75 & prob <= 1 ~ "High",
    TRUE ~ NA_character_
  ))

# Set factor levels to control legend order
Vinmonopolet_market$prob_category <- factor(Vinmonopolet_market$prob_category,
                                            levels = c("High", "Medium High", "Medium Low", "Low

merged_prob_data <- municipalities %>%
  left_join(Vinmonopolet_market, by = c("municip_code" = "mun_code")) %>%
  mutate(prob_category = replace_na(prob_category, "Has a store"))  # Fill NAs with "Has a

# Plot
ggplot(merged_prob_data) +
  geom_sf(aes(fill = prob_category), color = "white", size = 0.1) +
  scale_fill_manual(
    values = c(
      "High" = "darkgreen",
      "Medium High" = "#90EE90",
      "Medium Low" = "#FF6666",
      "Low" = "darkred",
      "Has a store" = "grey"
    ),
    name = "Probability"
  ) +
  coord_sf() +
  theme_minimal() +
  theme(
    plot.title = element_text(face = "bold", size = 14),
    axis.text.x = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks = element_blank()
  ) +
```

```
labs(title = "Predicted Probability by Municipality",
     x = NULL, y = NULL)
```