

Modelling Unemployment and Vaccination Rates

Damneet Thiara, 11170388 - STAT 447C

Introduction

By April 2020, the overall unemployment rate in Canada had hit 14.1%, an 8.4 percentage point increase from February 2020 (Statistics Canada, 2024). This unprecedented increase was just one month after COVID-19 was declared a pandemic by the World Health Organization (WHO) (Centers for Disease Control and Prevention, 2023). Since then, deaths have been unprecedented, alongside a dramatic downturn in economic activity. By December 9th, 2020, Health Canada had authorized the first COVID-19 vaccine: the Pfizer-BioNTech vaccine (Health Canada, 2020). Health Canada also reported that the Pfizer-BioNTech Comirnaty COVID-19 vaccine had shown to have been “95% effective in protecting trial participants from COVID-19 for those 16 years and older” (Health Canada, 2024).

With consumer demand and economic activity spiralling downwards amidst a succession of COVID-19 lockdowns and increasing case/death counts, it begs the question of how COVID-19 vaccines have affected these economic characteristics. Specifically, this paper aims to analyze the relationship between vaccination rates and unemployment rates in Canada from 2020 to 2023. We test the difference between a logarithmic Frequentist model versus a Bayesian model to predict Canadian unemployment rates based on vaccination rates, using time series data.

Literature Review

Previous studies on employing a Bayesian model to unemployment rates have been conducted by Datta et al (1999). By building a hierarchical Bayesian model, they use cross-sectional and time series data to estimate US state unemployment rates (p.1080). Additionally, Younes & Altug, (2021) use a Bayesian analysis to estimate the effects of COVID-19 on key economic indicators and find that unemployment increased, interest rates decreased, and prices fell after the first lockdown measures. These effects were then dampened or counteracted by stimulus packages. The increase in unemployment rates here ties in directly with the purpose of this project.

Loomba et al. conduct a Bayesian analysis on how vaccine misinformation affects vaccination rates, reporting that misinformation can decline intent to vaccinate by up to 6.4 percentage points (2021).

Finally, researchers have explored the relationship between vaccination rates and unemployment rates in many ways already. Hu et al. use an OLS model to find the relationship between vaccination rates and unemployment rates in the US. When analyzing domestic unemployment rates, they find a negative relationship between the two; an increase in vaccination rates is seen alongside a decrease in unemployment rates. However, when analyzing through a state-by-state view, they find that states with higher vaccination rates are more likely to have higher unemployment rates. Although this may seem counter-intuitive, the authors explain this by showing how states with higher vaccination rates experience a higher change in unemployment rates, indeed highlighting the benefits that higher vaccination rates have when it comes to unemployment (2022, p.196-198). Indeed, Roghani & Panahi (2021, p. 10) also find the same correlation between states with high unemployment rates and high vaccination rates. Similarly, Hansen & Mano (2023, p. 148) find that there is also a negative relationship between vaccination and unemployment rates, specifically using instrumental variable analysis to isolate the effects of vaccines by using pharmacy density as a proxy.

Notably, there is little to no literature on comparing Frequentist and Bayesian methods to estimate the relationship between vaccination rates and unemployment rates.

Analysis

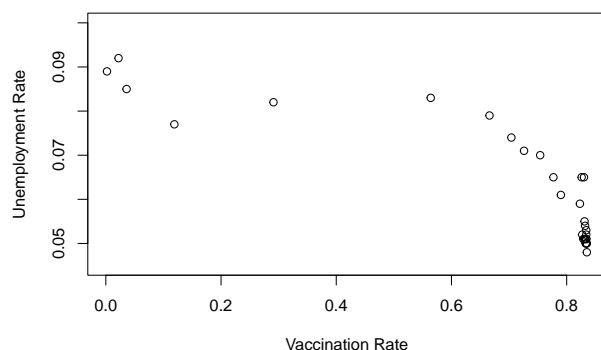
Data

Based on the literature review, although higher vaccination rates are associated with greater drops in unemployment, this does not necessarily mean that states or provinces with higher vaccination rates will

have lower unemployment rates. Since the purpose of this project is to test the differences between a Frequentist and Bayesian model, for simplicity, national time series data for Canadian unemployment rates and vaccination rates is used, rather than dividing it between multiple provinces.

The data for vaccination rates was retrieved from the Government of Canada Health Information Base (Public Health Agency of Canada, 2024). Previous literature uses one dosage of a COVID-19 vaccine as the benchmark for vaccination rates, so the same standard is used here. Canadian unemployment rates were retrieved from Statistics Canada databases (Statistics Canada, 2024). Unemployment rates are reported for an entire month, so the corresponding vaccination rate for each month is the last available vaccination rate available for the end of each month. Some months in 2023 are omitted due to limited vaccination rate data during this time. The final data looks like:

Vaccination and Unemployment Rates in Canada, 2020 to 2023



As vaccination rates go up, the unemployment rate that same month goes down.

Frequentist Model

To begin, we start by building a logarithmic Ordinary Least Squares model:

$$\log(U) = \beta_1 V + \beta_0$$

The fitted model is as follows:

```
##
## Call:
## lm(formula = log(ur) ~ v)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15709 -0.09830 -0.04948  0.11060  0.23772
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.36841    0.06106  -38.789  < 2e-16 ***
## v           -0.61204    0.08370   -7.312  7.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1243 on 27 degrees of freedom
## Multiple R-squared:  0.6645, Adjusted R-squared:  0.652
## F-statistic: 53.47 on 1 and 27 DF,  p-value: 7.265e-08
```

Bayesian Model

Considering the restrictions on unemployment and vaccination rates, being between 0 and 1, a Beta proportion distribution is appropriate. The Beta proportion regression model can be defined as:

$$U \sim \text{BetaProportion}(\mu, \sigma)$$

$$\mu \in (0, 1)$$

$$\sigma \in \mathbb{R}^+$$

$$U \in (0, 1)$$

We define μ using an inverse logit function, which takes all real values as inputs and values between 0 and 1 as the output.

$$\mu = \text{inv.logit}(\beta_1 V + \beta_0)$$

The first set of priors for the β_1 slope, β_0 intercept, and σ parameters are as follows:

$$\beta_1 \sim \text{Normal}(0, 10)$$

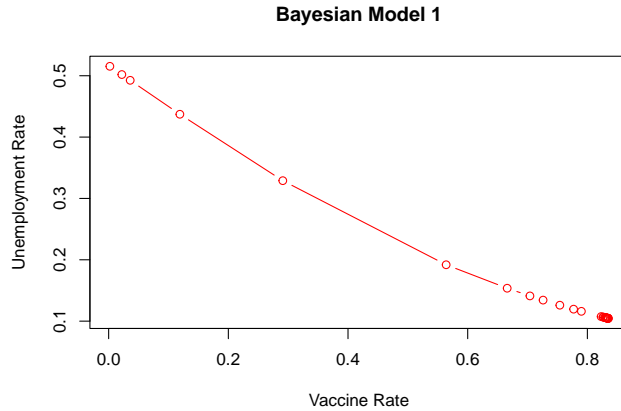
$$\beta_0 \sim \text{Exp}(0.1)$$

$$\sigma \sim \text{Exp}(0.01)$$

The slope parameter is normally distributed to allow for the possibility between positive and negative values. The intercept must be positive since we assume unemployment cannot be below 0, and σ must also be positive, hence the exponential distribution.

The preliminary model produces the following results:

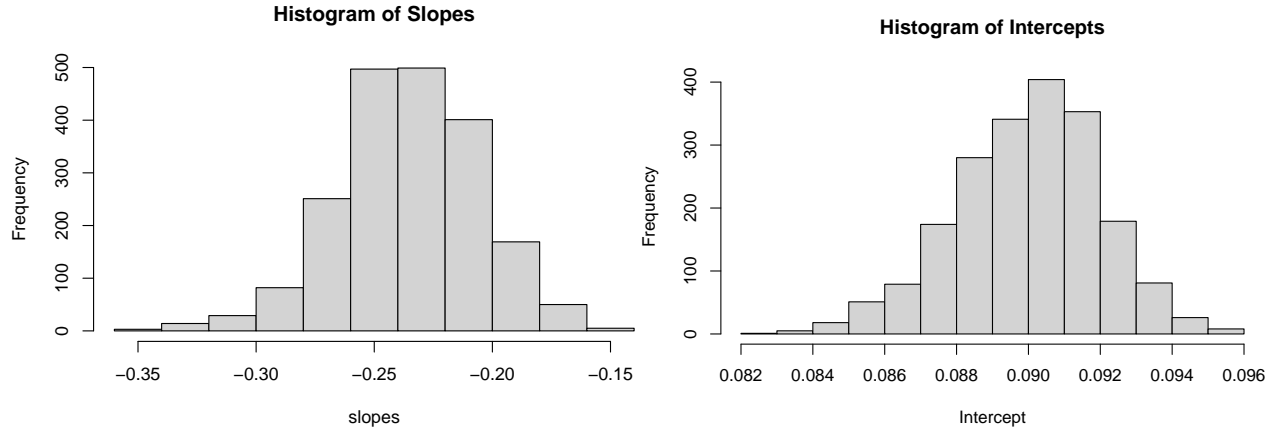
	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
slope	2.69	0.01	0.32	-3.29	-2.90	-2.69	-2.48	-2.06	1046	1
intercept	0.07	0.00	0.07	0.00	0.02	0.04	0.10	0.24	1118	1
sigma	6.15	0.05	1.75	3.25	4.93	5.96	7.18	10.13	1137	1



The unemployment rates are clearly inflated since the observed data for unemployment falls between 5% and 9%. The issue likely seems to be with the fact that the unemployment rate does not vary much and remains at low levels, typically below 10%. Employing a beta prior to this raw data might not be appropriate, since the beta distribution ranges between 0 and 1. To combat this, the unemployment rate is scaled upwards by a random constant. In this case, scaling upwards by 10 seems appropriate since none of the data for this model falls above 10%, so scaling by 10 will ensure that the data remains within the range of 0 to 1. After doing this, the following results are produced:

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
slope	-2.36	0.01	0.30	-2.99	-2.54	-2.35	-2.15	-1.79	743	1.00
intercept	2.21	0.01	0.23	1.78	2.05	2.21	2.36	2.70	737	1.00
sigma	45.59	0.38	11.47	25.87	37.59	44.66	52.40	70.90	893	1.01

The key observations here are the intercept and the negative slope. The histograms for both are shown below.



To test different priors, I change the model priors to the following:

$$\beta_1 \sim Normal(0, 1)$$

$$\beta_0 \sim Exp(0.01)$$

$$\sigma \sim Exp(0.1)$$

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
slope	-2.09	0.02	0.33	-2.74	-2.31	-2.08	-1.86	-1.45	460	1
intercept	2.02	0.01	0.25	1.54	1.84	2.01	2.19	2.51	473	1
sigma	34.09	0.31	9.29	18.06	27.55	33.06	40.02	54.46	881	1

The posterior estimates are slightly different but around the same range. Notably, the effective sample sizes have decreased in this new model, so the first model will be kept, considering the posterior estimates are only slightly different.

Conclusion

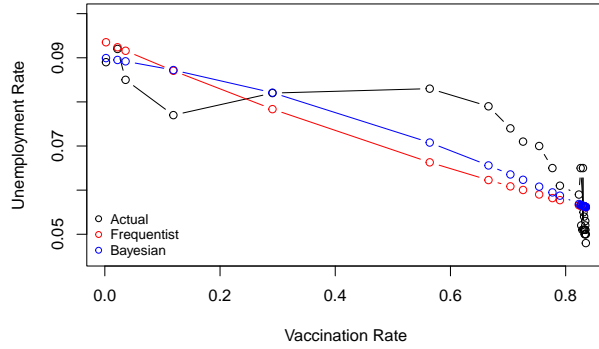
Results

The estimates for the parameters for each respective model are listed below:

	Frequentist	Bayesian
slopes	-0.61204	-0.2360000
intercepts	0.09366	0.0901144

Notably, the slopes are different, but both have different interpretations, so instead, the model data can be plotted to see how close it is to the observed. The model data for both the Frequentist model and the chosen Bayesian model are plotted against the actual data below:

Vaccination and Unemployment Rates in Canada, 2020 to 2023

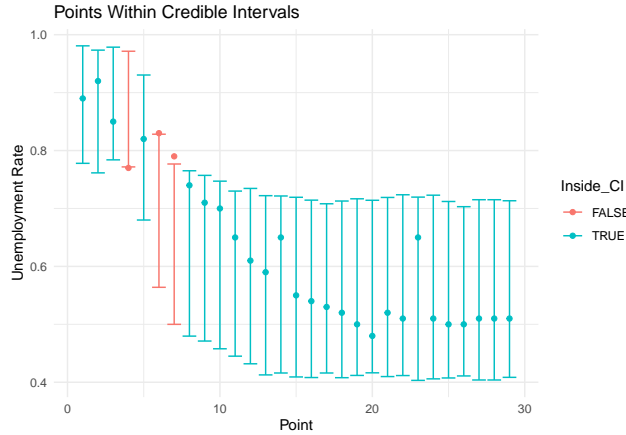


Both look similar, but the root mean squared error can be computed to compare the fit of both:

Frequentist	Bayesian
0.0077369	0.0066725

The RMSE for the Bayesian method is smaller, and thus fits better. It can safely be concluded that higher vaccination rates lead to lower unemployment rates, with an intercept of about 9% at 0% vaccination.

Finally, credible intervals and generated quantity predictions can be used to if the Bayesian model can predict the next missing data point. For each interval, we leave the i -th observation out:



Limitations

Frequentist The first limitation of this model is that the predicted unemployment rate could theoretically go above 1, which is not possible in the unemployment model. Data is also limited due to the short period in which unemployment was incredibly high and the vaccination rate was increasing, so the errors are likely higher than they could be if there was more data available. The usage of the Bayesian model aims to fix the first problem here.

Bayesian Although the Bayesian model ensures that results will not fall out of the appropriate range, given the range of the distributions, there are still limitations to this model. First, the data had to be scaled to be used with the Beta distribution, as was discussed earlier. Although this solved the issue in this particular case, if unemployment rates were to go above 10%, then scaling by 10 would cause the data to go outside the range of the Beta distribution. Thus, a different scalar constant would need to be chosen. This leads to an issue of the scalar being rather arbitrary. This issue does not arise in the Frequentist model.

References

- Centers for Disease Control and Prevention. (2023, March 15). CDC Museum COVID-19 timeline. <https://www.cdc.gov/museum/timeline/covid19.html>
- Datta, G. S., Lahiri, P., Maiti, T., & Lu, K. L. (1999). Hierarchical Bayes estimation of unemployment rates for the states of the U.S. *Journal of the American Statistical Association*, 94(448), 1074. <https://doi.org/10.2307/2669921>
- Hansen, N., & Mano, R. C. (2023). COVID-19 vaccines: A shot in arm for the economy. *IMF Economic Review*, 148-169. <https://doi.org/10.2139/ssrn.4026511>
- Health Canada. (2020, December 9). Health Canada authorizes first COVID-19 vaccine. <https://www.canada.ca/en/health-canada/news/2020/12/health-canada-authorizes-first-covid-19-vaccine0.html>
- Health Canada. (2024, March 28). Pfizer-biontech Comirnaty COVID-19 vaccine. <https://www.canada.ca/en/health-canada/services/drugs-health-products/covid19-industry/drugs-vaccines-treatments/vaccines/pfizer-biontech.html>
- Hu, F., Pang, J., & Sun, H. (2022). Research on the relationship between unemployment rate and vaccination rate in United States. *Journal of Economics, Business and Management*, 10(3), 195-199. <https://doi.org/10.18178/joebm.2022.10.3.698>
- Loomba, S., De Figueiredo, A., Piatek, S. J., De Graaf, K., & Larson, H. J. (2021). Measuring the impact of COVID-19 vaccine misinformation on vaccination intent in the UK and USA. *Nature Human Behaviour*, 5(3), 337-348. <https://doi.org/10.1038/s41562-021-01056-1>
- Public Health Agency of Canada. (2024, September 15). COVID-19 vaccination coverage in Canada. <https://health-infobase.canada.ca/covid-19/vaccination-coverage/>
- Roghani, A., & Panahi, S. (2021). Higher COVID-19 vaccination rates among unemployed in the United States: State level study in the first 100 days of vaccine initiation. *Division of Epidemiology - University of Utah*. <https://doi.org/10.1101/2021.04.17.21255668>
- Statistics Canada. (2024, April 5). Labour force characteristics by province, monthly, seasonally adjusted. <https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1410028703&pickMembers%5B0%5D=3.1&pickMembers%5B1%5D=4.1&cubeTimeFrame.startMonth=04&cubeT>
- Younes, O. A., & Altug, S. (2021). The COVID-19 shock: A Bayesian approach. *Journal of Risk and Financial Management*, 14(10), 495. <https://doi.org/10.3390/jrfm14100495>

Appendix

Cleaning Data

```
## Reading CSV files
vaccines<-read.csv("~/Desktop/vaccination-coverage-map_data.csv")
ur_data<-read.csv("~/Desktop/STAT 447C/PROJECT DATA/UR_DATA.csv")

## Isolating proportion at least 1 dose

Canadian_vaccines<-subset(vaccines, prename == "Canada")
Canadian_vaccines<-data.frame(
  date=Canadian_vaccines$week_end,
  dose_rate=Canadian_vaccines$proptotal_atleast1dose)

## Removing extra data

ur_data<-ur_data[-c(40,41),]
ur_data<-ur_data[-c(30:39),]

Canadian_vaccines<-Canadian_vaccines[-c(1,3:6,8:10,12:14,16:18,20:23,
                                         25:27,29:32,34:36,38:40,42:45,
                                         47:49,51:53,55:58,60:62,64:66,68:69,
                                         71,80,85:88),]

## Normalizing data

ur<-ur_data$UR/100
v<-Canadian_vaccines$dose_rate/100
N<-length(v)
df<-data.frame(Unemployment=ur,Vaccination=v)
```

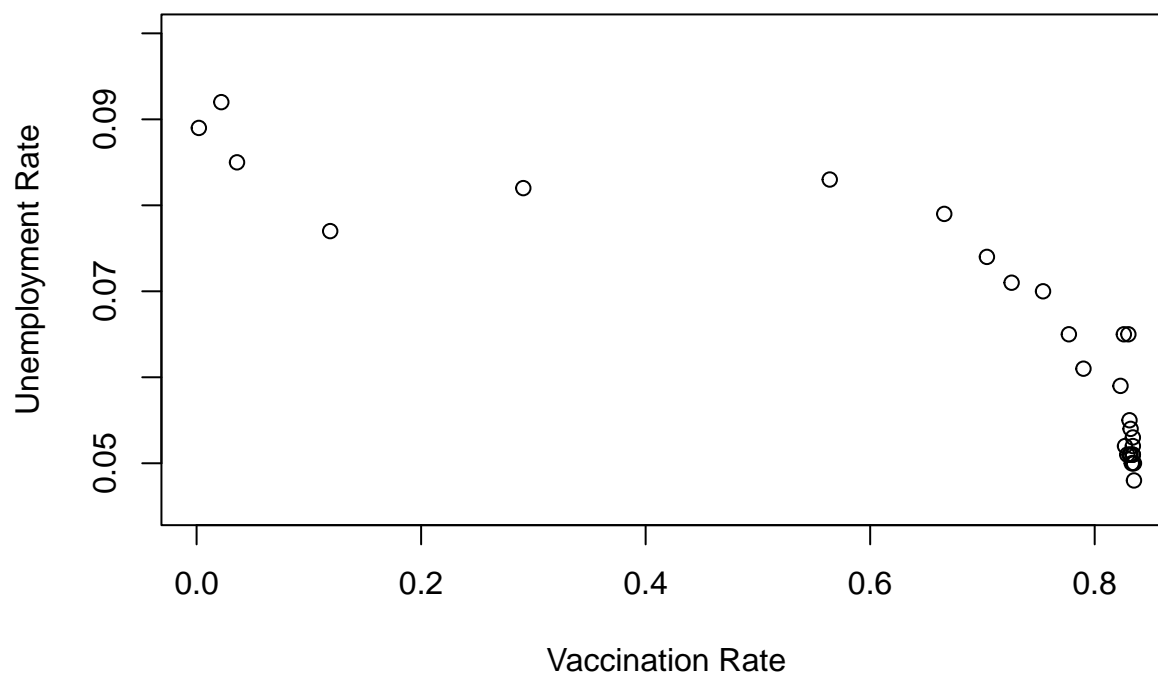
Table of Data

```
#rmarkdown::paged_table(df)

## Plot for first look at relationship between two variable

plot(v,ur,
     main="Vaccination and Unemployment Rates in Canada, 2020 to 2023",
     ylab="Unemployment Rate",
     xlab="Vaccination Rate",
     ylim=c(0.045,0.1))
```

Vaccination and Unemployment Rates in Canada, 2020 to 2023



Plot 1

Fitting Frequentist Model

```
## Logarithmic model
```

```
ordinary_model<-lm(log(ur)~v)
summary(ordinary_model)
```

```
##
## Call:
## lm(formula = log(ur) ~ v)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.15709 -0.09830 -0.04948  0.11060  0.23772
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.36841    0.06106  -38.789  < 2e-16 ***
## v           -0.61204    0.08370   -7.312  7.26e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1243 on 27 degrees of freedom
## Multiple R-squared:  0.6645, Adjusted R-squared:  0.652
## F-statistic: 53.47 on 1 and 27 DF,  p-value: 7.265e-08
```

Summary statistics for frequentist model.

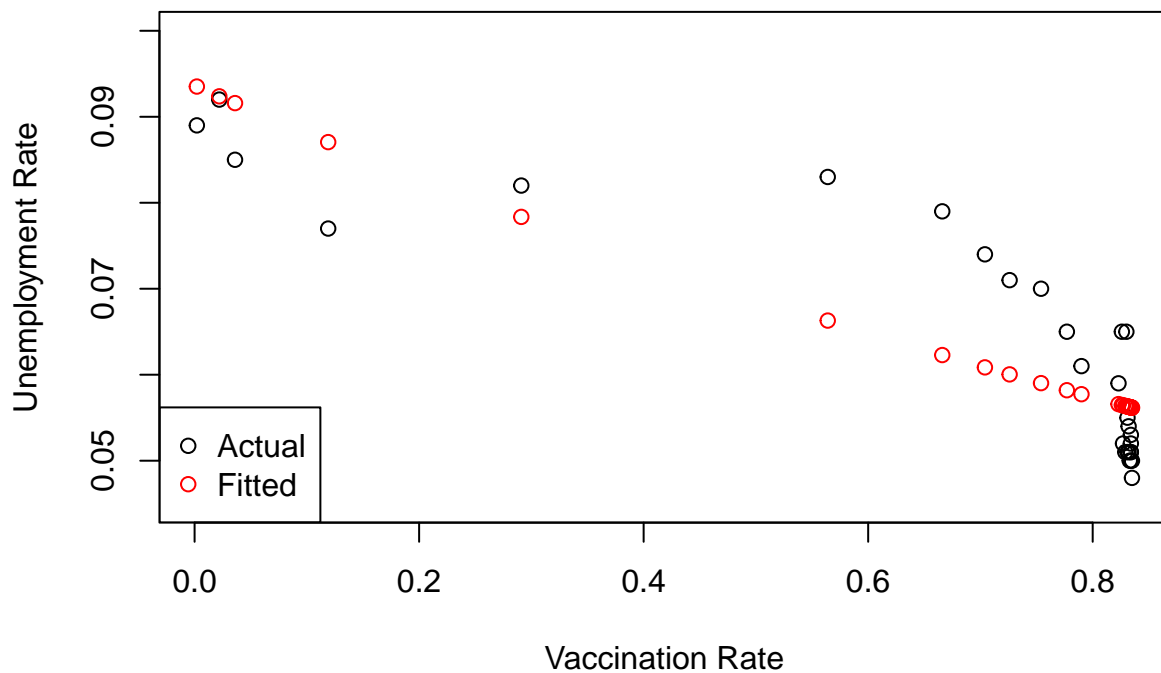

```
## Measuring accuracy

dosage<-data.frame(v)

predictions<-predict(ordinary_model,newdata=dosage)

plot(v,ur,
     main="Vaccination and Unemployment Rates in Canada, 2020 to 2023",
     ylab="Unemployment Rate",
     xlab="Vaccination Rate",
     ylim=c(0.045,0.1))
lines(dosage$v,exp(predictions),type="p",col="red")
legend("bottomleft", legend = c("Actual", "Fitted"),
      col = c("black", "red"), pch = c(1, 1))
```

Vaccination and Unemployment Rates in Canada, 2020 to 2023



```
errors_regular<-exp(predictions)-ur
sqrt(mean(errors_regular^2))
```

```
## [1] 0.007736867
```

Plot 2. Square root of the mean squared error of residuals for this model is 0.007736867.

Bayesian Model

Model 1 I use the following priors:

$$\begin{aligned}
 U &\sim \text{Beta}(\mu, \sigma) \\
 \mu &= \text{inv.logit}(\beta_1 V + \beta_0) \\
 \beta_1 &\sim \text{Normal}(0, 10) \\
 \beta_0 &\sim \text{Exp}(0.1)
 \end{aligned}$$

$$\sigma \sim \text{Exp}(0.01)$$

```
data {
  int<lower=0> N;
  vector<lower=0,upper=1>[N] v;
  vector<lower=0, upper=1>[N] u;
  real<lower=0,upper=1> v_pred;
}

parameters {
  real slope;
  real<lower=0> intercept;
  real<lower=0> sigma;
}

transformed parameters {
  vector[N] mu=inv_logit(intercept + slope*v);
}

model {
  slope ~ normal(0,10);
  intercept ~ exponential(0.1);
  sigma ~ exponential(0.01);
  u~beta_proportion(mu, sigma);
}

generated quantities {
  real u_pred = beta_proportion_rng(inv_logit(intercept + slope*v_pred),sigma);
}
```

```
fit1 = sampling(
  seed=123,
  MODEL1,
  data = list(u=ur,v=v,N=N,v_pred=1),
  chains = 4,
  iter = 1000
)
```

```
print(fit1)
```

Fitting Model

```
## Inference for Stan model: anon_model.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##           mean se_mean   sd  2.5%   25%   50%   75%  97.5% n_eff Rhat
## slope      -2.69     0.01  0.32  -3.29  -2.90  -2.69  -2.48  -2.06  1046 1.00
## intercept   0.07     0.00  0.07   0.00   0.02   0.04   0.10   0.24  1118 1.00
```

```

## sigma      6.15      0.05 1.75 3.25 4.93 5.96 7.18 10.13 1137 1.00
## mu[1]      0.52      0.00 0.02 0.50 0.50 0.51 0.52 0.56 1119 1.00
## mu[2]      0.50      0.00 0.02 0.48 0.49 0.50 0.51 0.55 1129 1.00
## mu[3]      0.49      0.00 0.02 0.47 0.48 0.49 0.50 0.54 1133 1.00
## mu[4]      0.44      0.00 0.02 0.41 0.42 0.43 0.45 0.48 1167 1.00
## mu[5]      0.33      0.00 0.02 0.29 0.31 0.33 0.34 0.38 1146 1.00
## mu[6]      0.19      0.00 0.03 0.14 0.17 0.19 0.21 0.26 1062 1.00
## mu[7]      0.15      0.00 0.03 0.11 0.13 0.15 0.17 0.22 1042 1.00
## mu[8]      0.14      0.00 0.03 0.10 0.12 0.14 0.16 0.20 1035 1.00
## mu[9]      0.13      0.00 0.03 0.09 0.11 0.13 0.15 0.20 1031 1.00
## mu[10]     0.13      0.00 0.03 0.08 0.11 0.12 0.14 0.19 1026 1.00
## mu[11]     0.12      0.00 0.03 0.08 0.10 0.12 0.14 0.18 1022 1.00
## mu[12]     0.12      0.00 0.03 0.07 0.10 0.11 0.13 0.18 1019 1.00
## mu[13]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.17 1014 1.00
## mu[14]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.16 1013 1.00
## mu[15]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.16 1013 1.00
## mu[16]     0.11      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[17]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[18]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[19]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[20]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[21]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.16 1013 1.00
## mu[22]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.16 1013 1.00
## mu[23]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.16 1013 1.00
## mu[24]     0.11      0.00 0.03 0.07 0.09 0.10 0.12 0.16 1013 1.00
## mu[25]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[26]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[27]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[28]     0.10      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## mu[29]     0.11      0.00 0.03 0.06 0.09 0.10 0.12 0.16 1012 1.00
## u_pred     0.07      0.00 0.10 0.00 0.01 0.03 0.09 0.34 2105 1.00
## lp_        26.99      0.05 1.33 23.46 26.42 27.32 27.95 28.55 688 1.01
##
## Samples were drawn using NUTS(diag_e) at Fri Apr 19 19:14:42 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

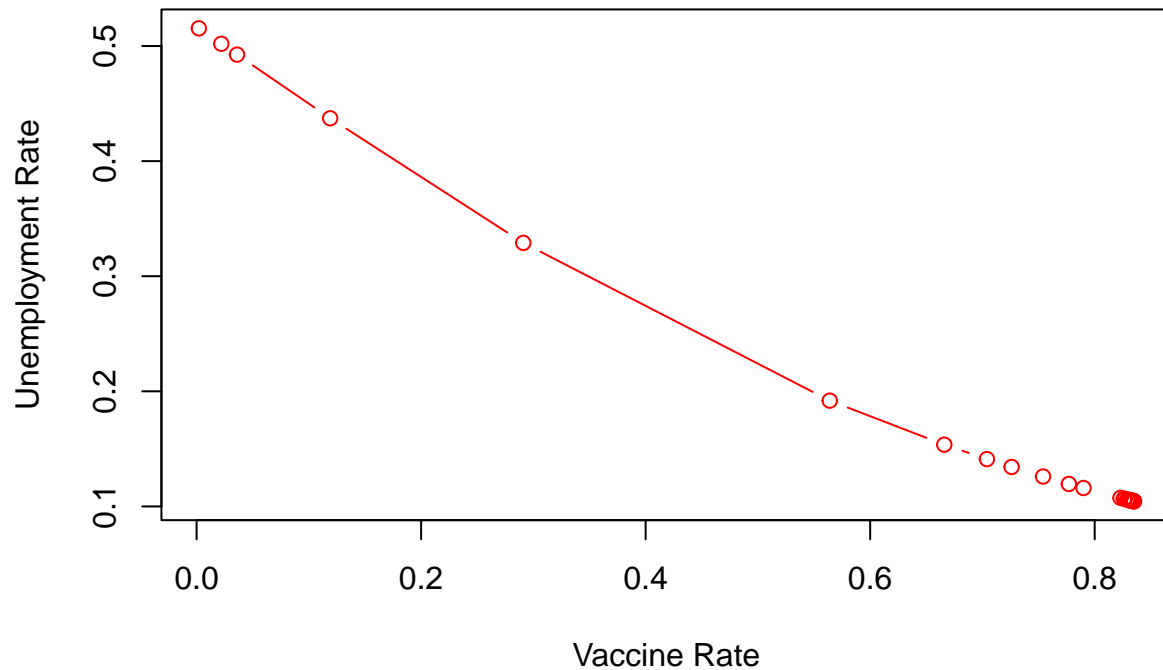
## Plot model

mu_values<-extract(fit1)$mu
averages <- colMeans(mu_values)

plot(v,averages,col="red",type="b",
     main="Bayesian Model 1",
     ylab="Unemployment Rate",
     xlab="Vaccine Rate")

```

Bayesian Model 1



Plot 3.

```
ur=ur_data$UR/10
```

```
fit2 = sampling(  
  seed=123,  
  MODEL1,  
  data = list(u=ur,v=v,N=N,v_pred=1),  
  chains = 4,  
  iter = 1000  
)
```

```
print(fit2)
```

Scaling Unemployment by 10

```
## Inference for Stan model: anon_model.  
## 4 chains, each with iter=1000; warmup=500; thin=1;  
## post-warmup draws per chain=500, total post-warmup draws=2000.  
##  
##
```

	mean	se_mean	sd	2.5%	25%	50%	75%	97.5%	n_eff	Rhat
## slope	-2.36	0.01	0.30	-2.99	-2.54	-2.35	-2.15	-1.79	743	1.00
## intercept	2.21	0.01	0.23	1.78	2.05	2.21	2.36	2.70	737	1.00
## sigma	45.59	0.38	11.47	25.87	37.59	44.66	52.40	70.90	893	1.01
## mu[1]	0.90	0.00	0.02	0.86	0.89	0.90	0.91	0.94	742	1.00
## mu[2]	0.90	0.00	0.02	0.85	0.88	0.90	0.91	0.93	743	1.00
## mu[3]	0.89	0.00	0.02	0.85	0.88	0.89	0.91	0.93	743	1.00
## mu[4]	0.87	0.00	0.02	0.83	0.86	0.87	0.89	0.91	748	1.00

```
## mu[5]      0.82    0.00  0.02  0.78  0.81  0.82  0.83  0.86   772 1.00
## mu[6]      0.71    0.00  0.02  0.67  0.70  0.71  0.72  0.74  1029 1.00
## mu[7]      0.66    0.00  0.01  0.63  0.65  0.66  0.67  0.68  1473 1.00
## mu[8]      0.64    0.00  0.01  0.61  0.63  0.64  0.65  0.66  1751 1.00
## mu[9]      0.62    0.00  0.01  0.59  0.61  0.62  0.63  0.65  1909 1.00
## mu[10]     0.61    0.00  0.01  0.58  0.60  0.61  0.62  0.64  2049 1.00
## mu[11]     0.59    0.00  0.02  0.56  0.59  0.59  0.60  0.62  2077 1.00
## mu[12]     0.59    0.00  0.02  0.56  0.58  0.59  0.60  0.62  2055 1.00
## mu[13]     0.57    0.00  0.02  0.54  0.56  0.57  0.58  0.60  1909 1.00
## mu[14]     0.57    0.00  0.02  0.53  0.56  0.57  0.58  0.60  1892 1.00
## mu[15]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.60  1862 1.00
## mu[16]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.60  1857 1.00
## mu[17]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1845 1.00
## mu[18]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1845 1.00
## mu[19]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1839 1.00
## mu[20]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1839 1.00
## mu[21]     0.57    0.00  0.02  0.53  0.56  0.57  0.58  0.60  1886 1.00
## mu[22]     0.57    0.00  0.02  0.53  0.55  0.56  0.58  0.60  1874 1.00
## mu[23]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.60  1868 1.00
## mu[24]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.60  1862 1.00
## mu[25]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1851 1.00
## mu[26]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1845 1.00
## mu[27]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1845 1.00
## mu[28]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.59  1845 1.00
## mu[29]     0.56    0.00  0.02  0.53  0.55  0.56  0.57  0.60  1857 1.00
## u_pred     0.47    0.00  0.08  0.31  0.41  0.46  0.52  0.62  1490 1.00
## lp__       39.64    0.04  1.21 36.63 39.07 39.93 40.58 41.06   759 1.00
##
```

```
## Samples were drawn using NUTS(diag_e) at Fri Apr 19 19:14:43 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

```
## Plotting model
```

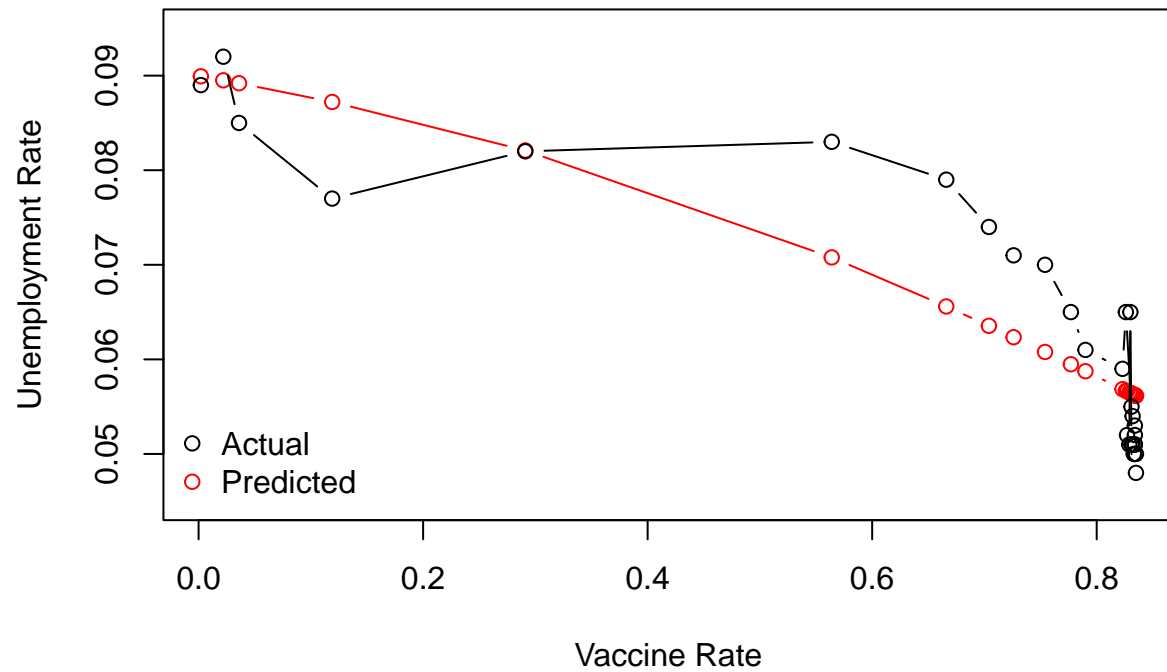
```
mu_values<-extract(fit2)$mu/10
averages <- colMeans(mu_values)
```

```
plot(v,averages,type="b",col="red",
     main="Bayesian Model 2 - Actual vs Predicted",
     ylim=c(0.045,0.095),
     xlab="Vaccine Rate",
     ylab="Unemployment Rate")
```

```
lines(v,ur/10,type="b")
```

```
legend("bottomleft", legend = c("Actual", "Predicted"), col = c("black", "red"), pch = c(1, 1), bty="n")
```

Bayesian Model 2 – Actual vs Predicted

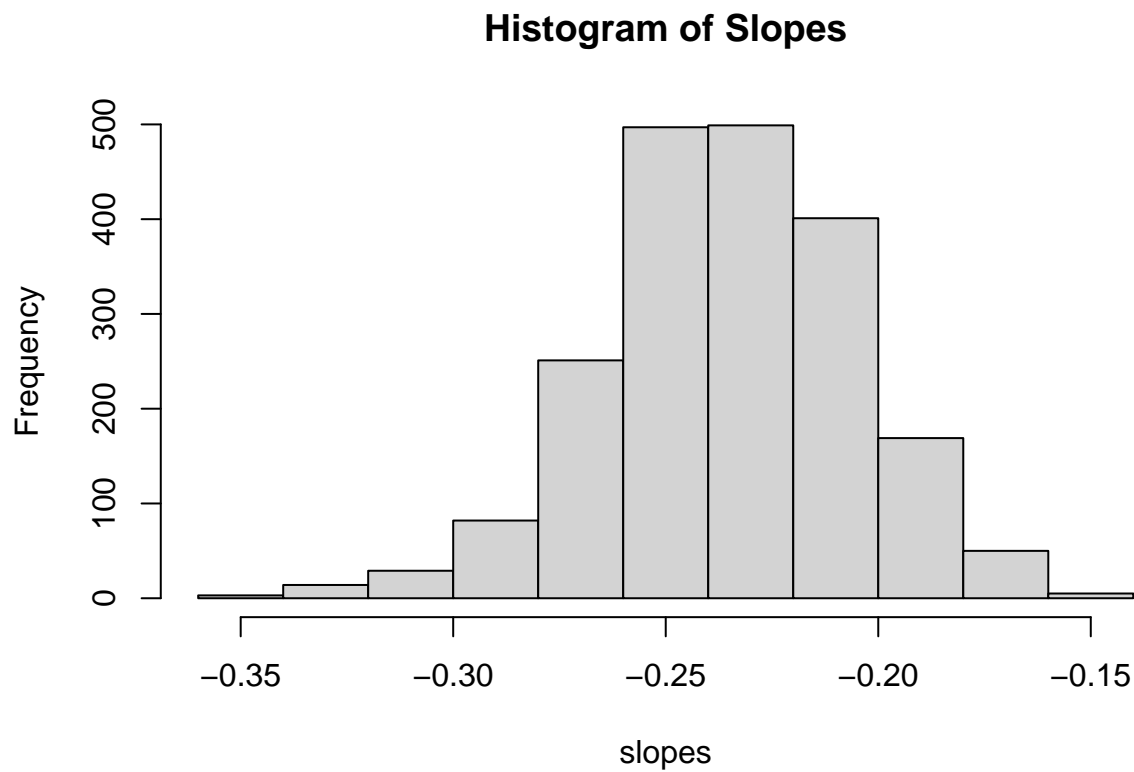


```
## Plotting slope and intercept histograms
```

```
slopes<-extract(fit2)$slope/10
```

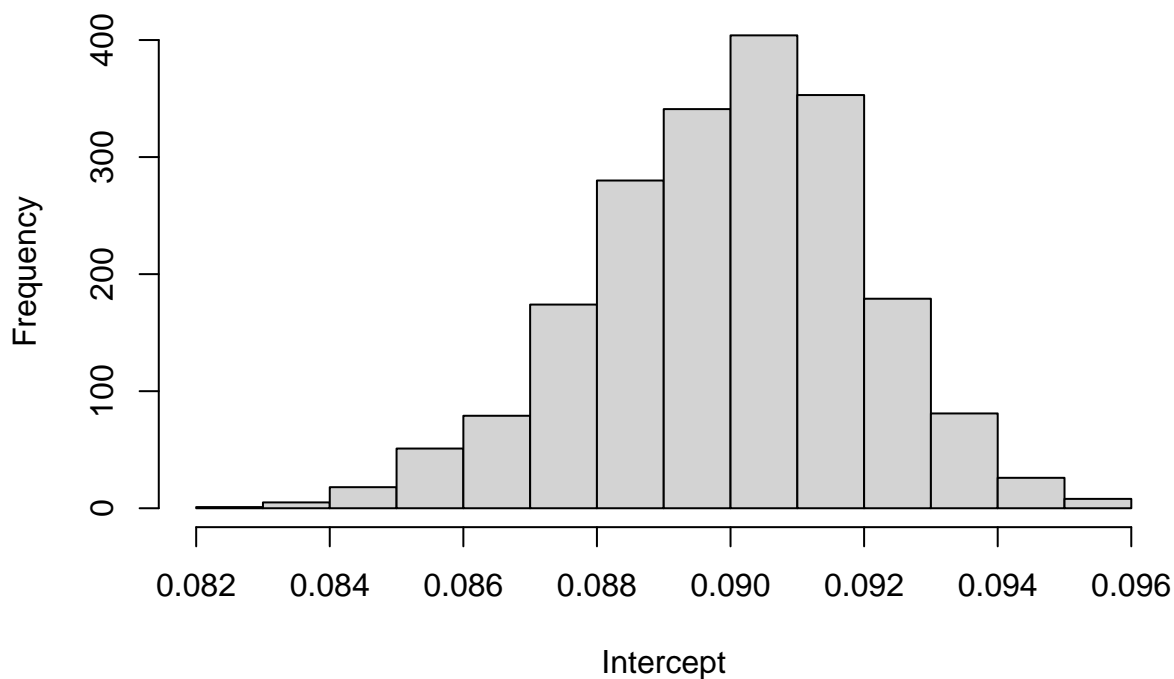
```
intercepts<-extract(fit2)$intercept
```

```
hist(slopes,main="Histogram of Slopes")
```



```
inv_logit <- function(x) {  
  exp_x <- exp(-x)  
  return(1 / (1 + exp_x))  
}  
  
hist(inv_logit(intercepts)/10,main="Histogram of Intercepts - Model 2",  
      xlab="Intercept")
```

Histogram of Intercepts – Model 2



Plot 4, 5, and 6.

```
errors_bayes1 <- data.frame(matrix(ncol = 29, nrow = 2000))
errors_bayes1<-averages-ur/10
sqrt(mean(errors_bayes1^2))
```

Errors

```
## [1] 0.006672511
```

Square root of the mean squared error for this model is 0.006672511.

Model 2: Changing Priors I use the following priors now:

$$\begin{aligned}
 U &\sim \text{Beta}(\mu, \sigma) \\
 \mu &= \text{inv.logit}(\beta_1 V + \beta_0) \\
 \beta_1 &\sim \text{Normal}(0, 1) \\
 \beta_0 &\sim \text{Exp}(0.01) \\
 \sigma &\sim \text{Exp}(0.1)
 \end{aligned}$$

```
data {
  int<lower=0> N;
  vector<lower=0,upper=1>[N] v;
  vector<lower=0, upper=1>[N] u;
  real<lower=0,upper=1> v_pred;
```



```

}

parameters {
  real slope;
  real<lower=0> intercept;
  real<lower=0> sigma;
}

transformed parameters {
  vector[N] mu=inv_logit(intercept + slope*v);
}

model {
  slope ~ normal(0,1);
  intercept ~ exponential(0.01);
  sigma ~ exponential(0.1);
  u~beta_proportion(mu, sigma);
}

generated quantities {
  real u_pred = beta_proportion_rng(inv_logit(intercept + slope*v_pred),sigma);
}

fit3 = sampling(
  seed=123,
  MODEL2,
  data = list(u=ur,v=v,N=N,v_pred=1),
  chains = 4,
  iter = 1000
)

print(fit3)

```

```

## Inference for Stan model: anon_model.
## 4 chains, each with iter=1000; warmup=500; thin=1;
## post-warmup draws per chain=500, total post-warmup draws=2000.
##
##               mean se_mean   sd  2.5%  25%   50%   75% 97.5% n_eff Rhat
## slope        -2.09    0.02 0.33 -2.74 -2.31 -2.08 -1.86 -1.45  460 1.00
## intercept     2.02    0.01 0.25  1.54  1.84  2.01  2.19  2.51  473 1.00
## sigma        34.09    0.31 9.29 18.06 27.55 33.06 40.02 54.46  881 1.00
## mu[1]         0.88    0.00 0.03  0.82  0.86  0.88  0.90  0.92  447 1.00
## mu[2]         0.88    0.00 0.03  0.82  0.86  0.88  0.89  0.92  449 1.00
## mu[3]         0.87    0.00 0.03  0.82  0.85  0.87  0.89  0.92  451 1.00
## mu[4]         0.85    0.00 0.03  0.80  0.83  0.85  0.87  0.90  460 1.00
## mu[5]         0.80    0.00 0.03  0.75  0.79  0.80  0.82  0.85  494 1.00
## mu[6]         0.70    0.00 0.02  0.66  0.69  0.70  0.71  0.73  802 1.00
## mu[7]         0.65    0.00 0.02  0.62  0.64  0.65  0.66  0.68 1462 1.00
## mu[8]         0.63    0.00 0.02  0.60  0.62  0.63  0.64  0.66 1865 1.00
## mu[9]         0.62    0.00 0.02  0.59  0.61  0.62  0.63  0.65 2070 1.00

```

```
## mu[10]      0.61      0.00 0.02 0.57 0.60 0.61 0.62 0.64 2172 1.00
## mu[11]      0.60      0.00 0.02 0.56 0.59 0.60 0.61 0.63 2083 1.00
## mu[12]      0.59      0.00 0.02 0.56 0.58 0.59 0.60 0.62 1979 1.00
## mu[13]      0.57      0.00 0.02 0.54 0.56 0.57 0.59 0.61 1552 1.00
## mu[14]      0.57      0.00 0.02 0.54 0.56 0.57 0.58 0.61 1516 1.00
## mu[15]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1465 1.00
## mu[16]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1455 1.00
## mu[17]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1436 1.00
## mu[18]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1436 1.00
## mu[19]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1426 1.00
## mu[20]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1426 1.00
## mu[21]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1505 1.00
## mu[22]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1485 1.00
## mu[23]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1475 1.00
## mu[24]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1465 1.00
## mu[25]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1446 1.00
## mu[26]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1436 1.00
## mu[27]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1436 1.00
## mu[28]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1436 1.00
## mu[29]      0.57      0.00 0.02 0.53 0.56 0.57 0.58 0.61 1455 1.00
## u_pred      0.49      0.00 0.09 0.30 0.42 0.49 0.55 0.67 2064 1.00
## lp__        33.44      0.06 1.35 29.93 32.78 33.81 34.44 35.00 551 1.01
##
## Samples were drawn using NUTS(diag_e) at Fri Apr 19 19:15:47 2024.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).
```

Errors

```
mu_values<-extract(fit3)$mu/10
averages <- colMeans(mu_values)

errors_bayes2<-averages-ur/10

sqrt(mean(errors_bayes2^2))
```

```
## [1] 0.006858772
```

This is higher than the square root mean squared error we had before, which was 0.006672511.

```
df<-data.frame(u=ur,v=v)

N_obs = nrow(df)
N_train = N_obs-1

ci_limits <- matrix(NA, nrow(df), 2)

for (i in 1:nrow(df)) {
  N_train <- nrow(df) - 1
  train_test_dta <- list(
    N = N_train,
```

```

    v = df$v[-i],
    u = df$u[-i],
    v_pred = df$v[i]
  )

  fit2 = sampling(
    seed=123,
    MODEL1,
    data = train_test_dta,
    chains = 4,
    iter = 1000
  )

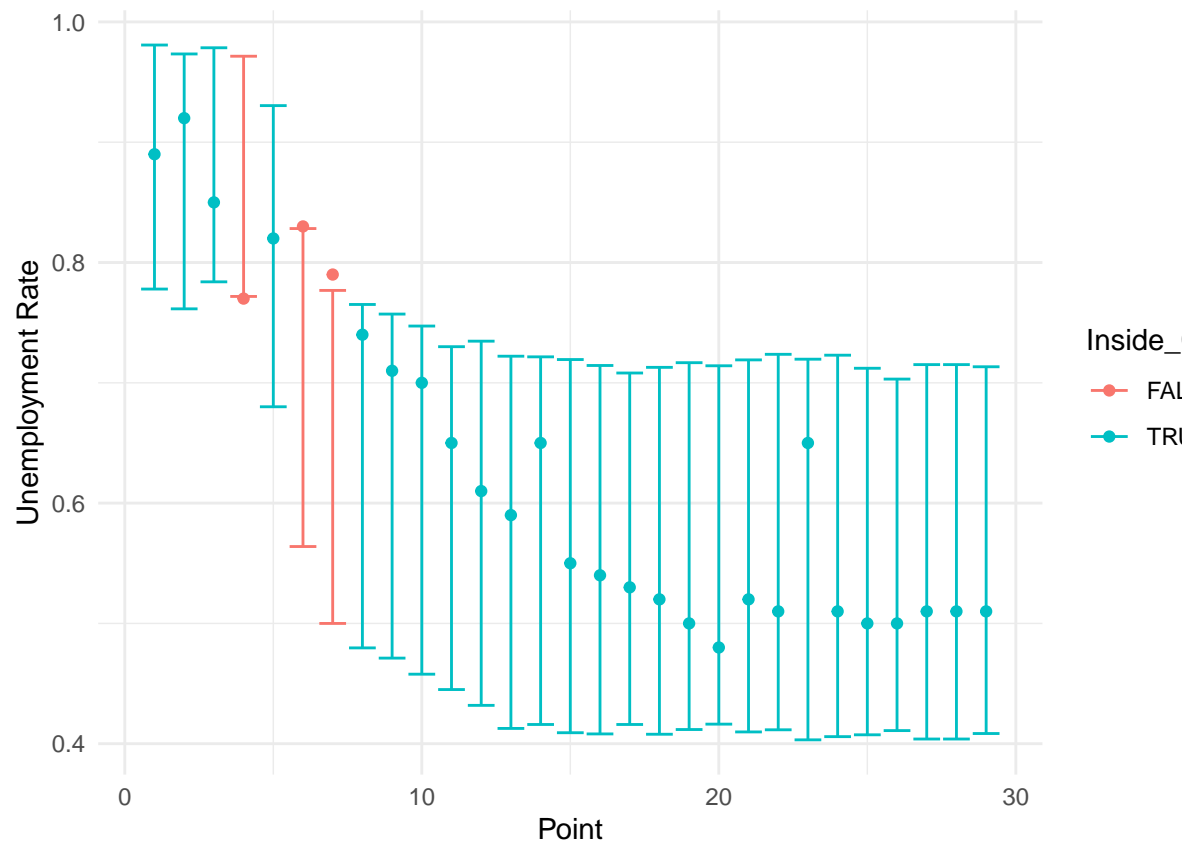
  samples <- (rstan::extract(fit2)$u_pred)

  obs_credible_interval <- quantile(samples, c(0.025, 0.975))

  ci_limits[i, ] <- obs_credible_interval
}

merged_df = df %>%
  bind_cols(data.frame(CI_L = ci_limits[,1], CI_R = ci_limits[,2])) %>%
  mutate(Inside_CI = (u >= CI_L & u <= CI_R))
merged_df %>%
  ggplot(aes(x = 1:N_obs, y = u, ymin = CI_L, ymax = CI_R, color=Inside_CI)) +
  geom_point() +
  geom_errorbar() +
  theme_minimal() +
  labs(x = "Point", y = "Unemployment Rate")

```



Credible Intervals

Plot 7. 95% credible intervals and predictions based on Model 1.