52525 Take-Home Final 2021

206096588

12 7 2021

The data I will analyze contains a database of new corona patients by localities in Israel in the two weeks 12/24/20 to 7/1/21. I will try to predict the rate of new cases (new cases / population).

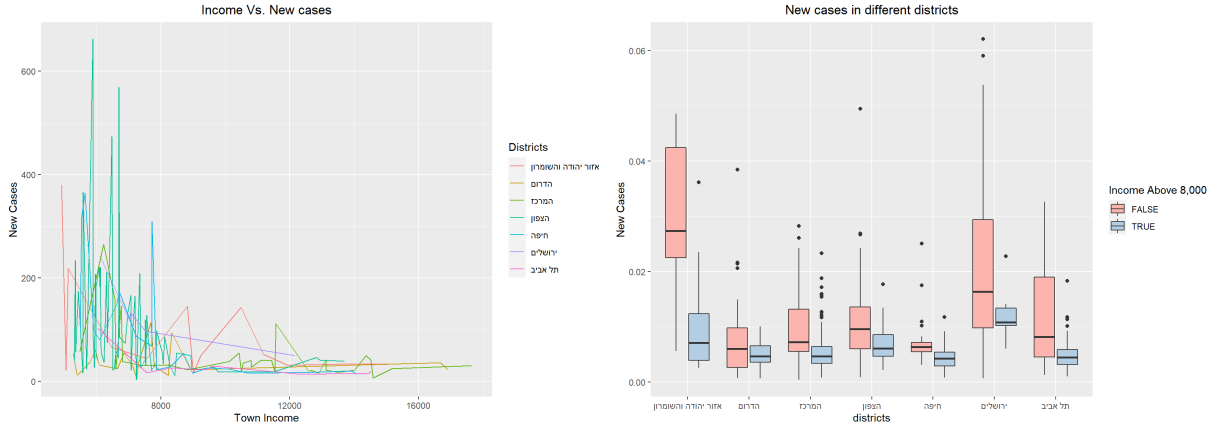
First I did a pre-processing and saw that there are missing values in the data. I completed these values with the help of the average of the column so that it does not hurt and affect the prediction too much. In addition, I examined the correlations between the various variables versus the response variable (new cases). I found that there are a number of variables that the correlation is very high in absolute value and some that the correlation is close to zero and later I will decide whether to remove these variables from the model.

The Correlation between the variables and the response variable

Vaccination	Bagrut	Income	Socioec_ind	pop_denisty	diabetes_rate	over20	over50	over70	perc_clalit	east_coord	north_coord
-0.227	-0.631	-0.418	-0.527	0.194	0.046	-0.088	-0.246	-0.326	-0.239	0.384	-0.064

Now, I will present two variables that show something interesting about the response variable and I have chosen to present them in two different graphs, which illustrates the conclusion I have reached.

Q-1



From the left graph (income Vs New cases) it can be seen quite clearly that when the income is lower so the number of new cases is higher. This may suggest that when income is lower there may be less opportunity to defend against the disease or there is a need to go out to work at any cost, or perhaps the type of work that does not allow working from home in low-wage jobs (not high-tech jobs for examples) so the number of infections increases accordingly.

From the right graph, (the boxplots) I decided to divide by districts and by jobs where the income is higher than 8000 versus lower than 8000. It can be seen quite clearly here too that when the salary is lower than 8000 (pink) in all the districts the number of infections is higher. We can also see that the median is higher in all the districts.

Both graphs lead to the same conclusion - the lower the salary in the same district, the higher the number of infections.

The upward correlation can also be seen which also shows a negative ratio as the above graphs show.

Q-2

A. In this question I have created the ridge function which returns the coefficients of the model.

```
# a

#ridge regression that input the train data and lambda and output the beta's:
ridge <- function(train_x, train_y, lambda){
  xTx <- as.matrix(t(train_x)) %*% as.matrix(train_x)
  beta <- solve(xTx + lambda*diag(ncol(train_x))) %*% as.matrix(t(train_x)) %*% train_y #(XtX+Lambda*I)-1*XtY - The formula
 of ridge regression
  return(beta)} #return the coefficiants
```

B. I then created the cv_ridge function which gets a list of lambdas and the train, first randomizes the data so that there is no particular order. Then classifies to 70% training and 30% validation. Performs the ridge function from above on the train and then checks the RMSE on the Validation. Then returns the coefficients of the model that had the lowest rmse, and the optimal lambda.

```
# b

#function that split the data into train and validation and find the optimal model
cv_ridge <- function(x, y, lambda, train_size = 0.7){
  set.seed(123)
  rmse_train <- c()
  rmse_train_build <- c()
  cases_samp <- x[sample(nrow(x), nrow(x), replace = FALSE),] #sampaling the data to get randomization choise to train and v
alidation.
  train_size_int <- as.integer(nrow(cases_samp)*train_size) #number of rows that are the train
  train_cases_sample <- cases_samp[c(1:train_size_int),] #the train sample
  validation_cases_sample <- cases_samp[c((train_size_int+1):nrow(cases_samp)),] #the validation sample

  for(i in 1:length(lambda)){
  beta_train <- ridge(train_cases_sample[,-ncol(train_cases_sample)],train_cases_sample[,ncol(train_cases_sample)], lambda
[i])
  rmse_train <- c(rmse_train, sqrt(mean(as.matrix(validation_cases_sample[,-ncol(validation_cases_sample)]) %*% beta_train
 - validation_cases_sample[,ncol(validation_cases_sample)])^2))} #the function to calculate the rmse
lamb_rmse <- data.frame("lambda" = lambda, "rmse" = rmse_train) #df of all the lambdas and rmases
optimal_lambda <- lamb_rmse[order(lamb_rmse$rmse),][1,1] #the first row (order data by mse)
model_rmse <- lamb_rmse[order(lamb_rmse$rmse),][1,2]
best_model <- ridge(train_cases_sample[,-ncol(train_cases_sample)], train_cases_sample[,ncol(train_cases_sample)], optimal_l
ambda) #the model with the optimal data

  return(list(best_model = round(best_model,3), model_rmse = round(model_rmse,3), model_lambda = optimal_lambda))}
```

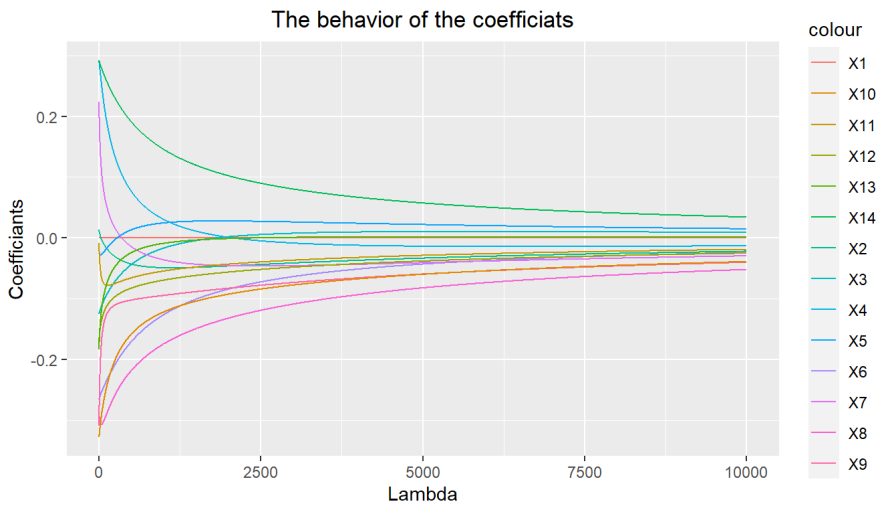
Now I want to see which variables are less important to me for the model and I can remove them. I decided to look at the correlations and thus decide which variables to remove. The variables with the lowest correlation are town_diabetes_rate, town_north_coord and pop_over20. Since I have a lot of variables left I do not get into a state of overfitting so I choose to remove all three. In addition, I scale the data so that the data would be of the same order of magnitude and added the intercept

Below is the optimal model I got with the minimum rmse and the lambda 0.0001.

```
## $best_model
##                                     [,1]
## X                                  0.013
## town_code                         -0.008
## agas_code                         -0.135
## accumulated_vaccination_first_dose  0.312
## town_pop_denisty                  -0.006
## town_perc_clalit                  -0.245
## town_income                        0.247
## town_bagrut                       -0.261
## town_socioeconomic_index          -0.351
## agas_socioeconomic_index          -0.346
## pop_over50                        -0.024
## pop_over70                        -0.164
## population                        -0.194
## town_east_coord                    0.295
##
## $model_rmse
## [1] 0.045
##
## $model_lambda
## [1] 0.0001
```

As we have learned, ridge regression is a regression that similar to estimating OLS only that it receives a certain penalty (lambda). In the given model and data, as it seems there is no need for a large fine in order to get a low rmse.

In addition I wanted to check the behavior of my coefficients, and see that indeed as the lambda grows so they approach zero and it can be seen that in the coefficient model they do behave this way.



C. In this section we will look at the residuals of the model (y-y_hat) and examine their behavior.

I decided to present two graphs that show the behavior of the residues in the model most clearly, and see if the residues are normally distributed

```
#c

#y-y_hat to find the residuals
residuals_vec <- data.frame("residuals" = train_cases_fit[,ncol(train_cases_fit)] - as.matrix(train_cases_fit[,-ncol(train_c
ases_fit)]) %*% ridge_regg$best_model)

#ploting the residuals plots:
p1 <- ggplot(residuals_vec, aes(residuals)) + geom_histogram(alpha=0.3, fill='indianred3', colour='black', binwidth=.1) + la
bs(title = "Residuals Histogram") + xlab("Residuals") + theme(plot.title = element_text(hjust = 0.5), plot.subtitle = elemen
t_text(hjust = 0.5))

resid_df <- data.frame("residuals" = residuals_vec$residuals, "real_y" = train_cases_fit[,ncol(train_cases_fit)]) #df of the
residuals and the real Y
p2 <- ggplot(resid_df) + geom_point(aes(real_y, residuals ),col = 'indianred3') + geom_abline() + labs(title = "Residuals Vs
Real Y") + xlab("Real Y") + ylab("Residuals") + theme(plot.title = element_text(hjust = 0.5), plot.subtitle = element_text(h
just = 0.5))

grid.arrange(p1, p2, ncol=2)
```

The histogram on the right shows the residuals and you can see the shape of a bell and that the residuals are indeed approaching almost normal distribution. You can see that there is a right tail but most of the residue is around zero.

In the right graph I chose to show the remnants in front of the real Y to see the connection between them. It can be seen that the relationship is reminiscent of a linear relationship but there is a large concentration around the 0.

I checked whether the residues are correlative to the various variables and it can be seen in the table below that the correlation tends to 0 in each of the variables.

```
#checking the residual and the variables correlations:
residuals <- residuals_vec$residuals
corr_table_resid <- data.frame(Vaccination = round(cor(train_cases$accumulated_vaccination_first_dose,residuals),3), Bagrut
 = round(cor(train_cases$town_bagrut ,residuals),3), Income = round(cor(train_cases$town_income ,residuals),3), Socioec_ind
 = round(cor(train_cases$town_socioeconomic_index, residuals),3), pop_denisty = round(cor(train_cases$town_pop_denisty ,resi
duals),3), diabetes_rate = round(cor(train_cases$town_diabetes_rate ,residuals),3), over20 = round(cor(train_cases$pop_over2
0, residuals),3), over50 = round(cor(train_cases$pop_over50, residuals),3), over70 = round(cor(train_cases$pop_over70, resid
uals),3), perc_clalit = round(cor(train_cases$town_perc_clalit, residuals),3), east_coord = round(cor(train_cases$town_east_
coord, residuals),3), north_coord = round(cor(train_cases$town_north_coord, residuals),3) )

kbl(corr_table_resid, caption = "The Correlation between the variables and the Residuals")  %>%   kable_paper("hover", full_
width = F) %>% kable_classic(html_font = "Cambria")
```

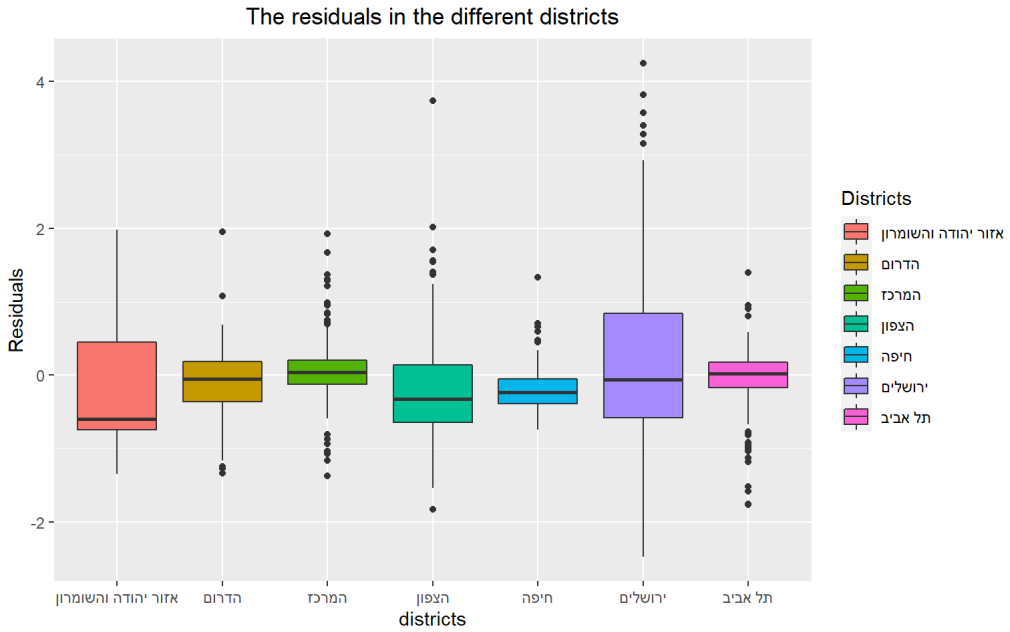
The Correlation between the variables and the Residuals

Vaccination	Bagrut	Income	Socioec_ind	pop_denisty	diabetes_rate	over20	over50	over70	perc_clalit	east_coord	north_coord
0.006	0.01	0.022	0.023	0.001	-0.077	0.001	0.008	0.012	-0.028	-0.016	-0.064

I will now present the relationship between one of the variables (district) versus the residuals. I chose to display it in a boxplot because the variable is categorical variable.

```
resid_df$mahoz <- train_cases$mahoz

ggplot(resid_df)+ geom_boxplot(aes(x = mahoz ,y = residuals, fill = mahoz )) + labs(title = "The residuals in the different
 districts", fill="Districts") + xlab("districts") + ylab("Residuals") + theme(plot.title = element_text(hjust = 0.5), plot.
subtitle = element_text(hjust = 0.5))
```



It can be seen that the median of the residues in each of the districts is quite close and around zero. The residues in the Yehuda and Shomron area are relatively lower than the rest. It can be seen that in Jerusalem and North there are exceptional residues that are relatively higher than the rest of the residues. It can be understood from this that the quality of the prediction in the districts of Jerusalem and Yehoda and Shomron and also the North is relatively less good in part of the towns.

D. Next I checked for one specific observation shown below, which are the 5 observations that have the biggest impact on it. I took the weight of the Ridge Regression X(XtX + Lambda I) -1 Xt And I checked which 5 observations (lines) get the maximun in this matrix.

```
#d

#choose one response observation:
response_observe <- train_cases %>% filter(agas_code==1)
response_observe <- response_observe[1,]

#droping the non correlation variable from the response observation:
droping <- c("X", "town_diabetes_rate","town_north_coord", "town_eng.y")
response_observe <- response_observe[,!(names(response_observe) %in% droping)]
#changing names of the columns:
response_observe <- response_observe %>% rename("town_c" = town_code, "agas_c" = agas_code , "vaccination_first_dose" = accu
mulated_vaccination_first_dose, "pop_denisty" = town_pop_denisty , "perc_clalit" = town_perc_clalit , "income" = town_incom
e, "bagrut" = town_bagrut, "town_socioec_ind" = town_socioeconomic_index, "agas_socioec_ind" = agas_socioeconomic_index)

train_cases_fit <- data.frame(scale(train_cases_fit)) #scaling the data
train_cases_fit$X <- 1 #adding the intercept


#calculate the weight of the regression:
xTx <- as.matrix(t(train_cases_fit)) %*% as.matrix(train_cases_fit)
xTxX <- solve(xTx + 0.001*diag(ncol(train_cases_fit))) %*% as.matrix(t(train_cases_fit))
XxtxX <- data.frame(as.matrix(train_cases_fit) %*% xTxX)

#choose the 5 most impact on the observation above (line number 2)
response_observe_W <- order(XxtxX[,2])[1:5]
train_best_predicted <- train_cases[response_observe_W,]

#droping the non correlation variables:
droping <- c("X", "town_diabetes_rate","town_north_coord", "town_eng.y")
train_best_predicted <- train_best_predicted[,!(names(train_best_predicted) %in% droping)]
#changing names of the columns:
train_best_predicted <- train_best_predicted %>% rename("town_c" = town_code, "agas_c" = agas_code , "vaccination_first_dos
e" = accumulated_vaccination_first_dose, "pop_denisty" = town_pop_denisty , "perc_clalit" = town_perc_clalit , "income" = to
wn_income, "bagrut" = town_bagrut, "town_socioec_ind" = town_socioeconomic_index, "agas_socioec_ind" = agas_socioeconomic_in
dex)

#the tables:
kbl(train_best_predicted, caption = "The observations which had the greatest impact on the forecast")  %>%   kable_paper("ho
ver", full_width = F) %>% kable_classic(html_font = "Cambria")
```

The observations which had the greatest impact on the forecast

	town_c	agas_c	town	vaccination_first_dose	mahoz	pop_denisty	perc_clalit	income	bagrut	town_socioec_ind	agas_socioec_ind	pop_over20
488	4502	0	עין קנייא	509	הצפון	6315.897	33.9	5987.091	82.00000	-1.030	0.1445719	1335
486	4203	0	מסעדה	406	הצפון	6315.897	57.5	5360.886	86.30137	-1.023	0.1445719	2340
483	4001	0	בוקעאתא	787	הצפון	334.100	54.9	5690.178	85.45455	-1.133	0.1445719	4292

	town_c	agas_c	town	vaccination_first_dose	mahoz	pop_denisty	perc_clalit	income	bagrut	town_socioec_ind	agas_socioec_ind	pop_over20
485	4201	0	מג'דל שמס	1680	הצפון	718.000	38.6	5341.246	72.28261	-0.800	0.1445719	7401
461	3650	0	אפרת	2697	אזור יהודה והשומרון	6315.897	22.9	10484.862	84.72222	0.425	0.1445719	5814

```
kbl(response_observe, caption = "The response observation")  %>%   kable_paper("hover", full_width = F) %>% kable_classic(ht
ml_font = "Cambria")
```

The response observation

town_c	agas_c	town	vaccination_first_dose	mahoz	pop_denisty	perc_clalit	income	bagrut	town_socioec_ind	agas_socioec_ind	pop_over20	pop_over
31	1	אופקים	955	הדרום	2796.3	66.4	7123.442	45.19481	-0.698	-0.5692505	3661	15

My chosen observation is the city Ofakim, and you can see the data of the observation above It can be seen that the 5 most influential cities are not so similar to the explained observation but there are similar to each other and it is interesting (and they are all from agas 0).

Q-3

```
#function that use the random forrest classifier and return the prediction with the test data
rf_prediction_model <- function(train_data, test_data){

random_forest_classifier <- randomForest(as.matrix(train_data[,-ncol(train_data)]), y=factor(train_data[,ncol(train_data)]),
keep.forest = TRUE) #build function of random forest

  return(predict(random_forest_classifier, test_data))}

#remove variables that are non-numeric and low correlated:
drop_not_cor <- c("X", "town_diabetes_rate","town_north_coord", "town", "town_eng.y", "mahoz")
train_cases_rf <- train_cases[,!(names(train_cases) %in% drop_not_cor)]
test_cases_rf <- test_response[,!(names(test_response) %in% drop_not_cor)]


#the prediction to the new cases rate in the test data:
predict_y <- as.numeric(as.character(rf_prediction_model(train_cases_rf, as.matrix(test_cases_rf))))

#the prediction to the rmse in the train data:
predict_rmse <- sqrt(mean((train_cases_rf$new_cases_rate - predict_y)**2))

cat("The predict rmse of my model is:",round(predict_rmse,4))
```

```
## The predict rmse of my model is: 0.0101
```

```
#saving my prediction:
save(predict_y, predict_rmse, file = "206096588.rda")
```

From looking at the data and getting acquainted with the different prediction methods we learned in the course, I decided to choose the Random Forest prediction method, because as I also saw in the lab, it is effective and predicts very well. This method basically create a data boostrap with replace, selects a limited number of variables to test and creates a tree and predicts it. Finally finds the average of the predictions to get the most accurate prediction.

Most of the variables are correlated to the explainatry variable (new cases rate) except of the 3 I presented above so I decided to remove them here as well.

I tried to make transformations on the data but it did not significantly affect the rmse and the predication so I decided to leave the variables as they are.

In the model I created I got a very small rmse which means it is the percentage error of the model, which is better than the Ridge model I performed earlier and other models I tested.