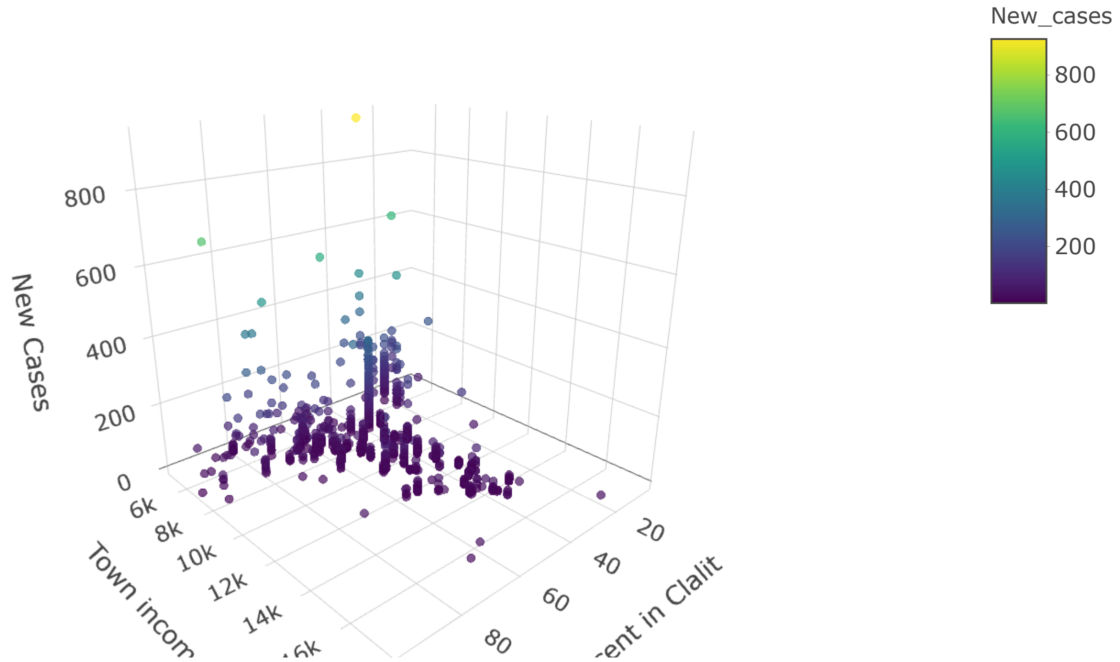
Final Exam

206094278

12 7 2021

1. Answer:

New cases as a function of % of Clalit members and Town income



```
## [1] "Correlation between new cases and pecent of clalit members -0.118"
```

```
## [1] "Correlation between new cases and town income -0.396"
```

```
## [1] "Correlation between pecent of clalit members and town income -0.171"
```

In the plot we can see that for both variables the number of new cases drops when there is an increase in both variables(they even have negative correlation with the new cases variable).
However, As the town income rises the percent of members in clalit drops so we have two variables that have negative correlation with new cases but do not co-exist(have negative correlation with each other). This can be may explained Clalit wide distribution of branches across the country which means services can be transferd to faster and it very popular in the periphery (which can explain it's high percent in low income areas). But, as the income rises members prefer premium services and leave clalit for a better health services. sorce[https://www.calcalist.co.il/local/articles/0,7340,L-3896539,00.html (https://www.calcalist.co.il/local/articles/0,7340,L-3896539,00.html)] All in all, clalit is more accessible to pepole and does give essential services but if pepole prefer leave it for better services if they have high income.

2.

   a.

```
ridge <- function(train_x, train_y, lambda){
  XtX <- t(as.matrix(train_x))%*%as.matrix(train_x)
  beta <- solve(XtX + diag(lambda,dim(XtX)[2]))%*%t(as.matrix(train_x))%*%as.matrix(train_y)
 return((beta))
}
```

b.

```
set.seed(3)

cv_ridge <- function(x, y, lambda = NA, train_size = 0.7){
  all_data <- cbind(x,y) # bind the data
  agas_code0_train <- all_data %>% filter(agas_code==0) %>%  sample_frac(train_size) # sample from the small cities
  agas_code0_test <- anti_join(all_data %>% filter(agas_code==0),agas_code0_train,by="town_code")
  agas_code_pos_train <- all_data %>% filter(agas_code!=0) %>% group_by(town_code) %>%  sample_frac(train_size) # sample from the large cities such that from each  # city we will sample number agas proportional to the city
  agas_code_pos_test <- anti_join(all_data %>%filter(agas_code!=0),agas_code_pos_train,by=c("town_code","agas_code"))
  train_x <-as.data.frame(rbind(as.data.frame(agas_code0_train),as.data.frame(agas_code_pos_train)))
  test_x <- as.data.frame(rbind(as.data.frame(agas_code0_test),as.data.frame(agas_code_pos_test)))
  train_y <- train_x[,dim(train_x)[2]] # split the data to x and y
  test_y <- test_x[,dim(test_x)[2]]
  train_x <- train_x[,-dim(train_x)[2]]  %>% select(-town_code,-agas_code)
  test_x <- test_x[,-dim(test_x)[2]]  %>% select(-town_code,-agas_code) # drop the # columns that used for the split(we can not calculate regression on them)

  betas <- c()
  mses <- c()
  for (i in 1:length(lambda)) {
    betas <- cbind(betas,ridge(train_x,train_y,lambda[i]))
    mses <- c(mses,mean((as.matrix(test_x)%*%as.vector(betas[,i])-test_y)^2))
  }
  model_mse <- round(min(mses),5)
  best_model <- round(as.vector(betas[,which(mses==min(mses))]),7)
  model_lambda <-round(lambda[which(mses==min(mses))],5)

  return(list(best_model = best_model, model_mse = model_mse, model_lambda = model_lambda))
}
x <- data %>% select(-new_cases,-town,-mahoz,-town_eng.y,-town_pop_denisty,-town_diabetes_rate,-agas_socioeconomic_index,-town_north_coord,-town_east_coord,-population)
# to have a better fit i decided to scale some variables in the data by total poplulation to have the variable %
x$accumulated_vaccination_first_dose <-100*x$accumulated_vaccination_first_dose/data$population
x$pop_over20 <-100*x$pop_over20/data$population
x$pop_over50 <-100*x$pop_over50/data$population
x$pop_over70 <-100*x$pop_over70/data$population

x <- cbind(1,x)
y <- data %>% select(new_cases)
y <- y/data$population
lambda <- seq(0,5,length.out = 1000)
model1<-cv_ridge(x,y,lambda,0.7)
model1
```

```
## $best_model
## [1]  0.0480257  0.0000675 -0.0000507 -0.0000002 -0.0001744 -0.0002645 -0.0004716
## [8]  0.0002862 -0.0001499
## 
## $model_mse
## [1] 0.00004
## 
## $model_lambda
## [1] 0.03003
```

Explaination

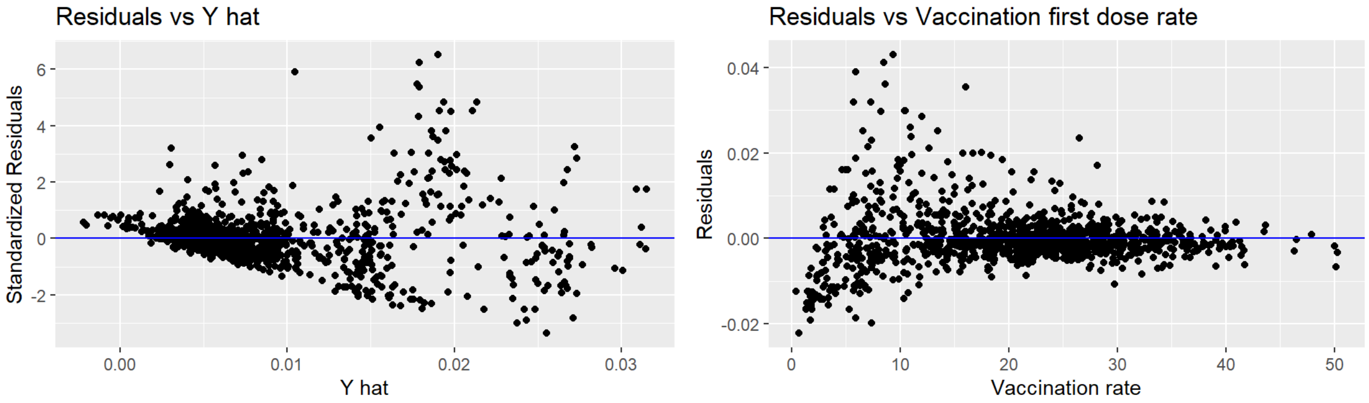
First, i wrote a in the function a part the splits the data to train and validation sets.
in order to avoid cases were a small amount of cities are sampled because they have more geographic statistical zones to sample from. I divided the data to cities that have agas_code==0 and agas_code!=0 and sampled a train_size data set for the cities that have agas_code==0 and a train_size from each citiy with agas_code!=0 (for example,if Tel Aviv has 10 agas and train_size = 0.7 i will sample 7 agas from Tel aviv). Later,the function calculate ridge regression over a range of lambdas and chooses the model with the min mse. As, for the variables in the regression i chose only the variables with rational scale such as bagrut rate and drop variables with NA in there rows(i decided to drop the variables instead of observations because dropping observation may hurt the external validity of the model). Also, i decided to scale variables in the data by total poplulation

c.

```
# first calculate the residoals
for_cal <- x %>% select(-town_code,-agas_code)
bet <- ridge(for_cal,y,lambda = model1$model_lambda)
y_hat <-  as.matrix(for_cal)%*%as.vector(bet)
res <- y-y_hat
n_ress <- (res$new_cases - mean(res$new_cases))/sd(res$new_cases)
n_ress <- as.data.frame(n_ress)
```

Answer

```
dd <- as.data.frame(cbind(y_hat,n_ress$n_ress))
dd<-dd %>% ggplot(aes(x=V1,y=V2))+ geom_point() + ggtitle("Residuals vs Y hat") +xlab("Y hat") + ylab(
"Standardized Residuals") + geom_hline(yintercept=0,col="blue")
dp <- as.data.frame(cbind(for_cal$accumulated_vaccination_first_dose,res$new_cases))
dp<-dp %>% ggplot(aes(x=V1,y=V2)) + geom_point()+ggtitle("Residuals vs Vaccination first dose rate") +
xlab("Vaccination rate") + ylab("Residuals") + geom_hline(yintercept=0,col="blue")
ggarrange(dd,dp)
```



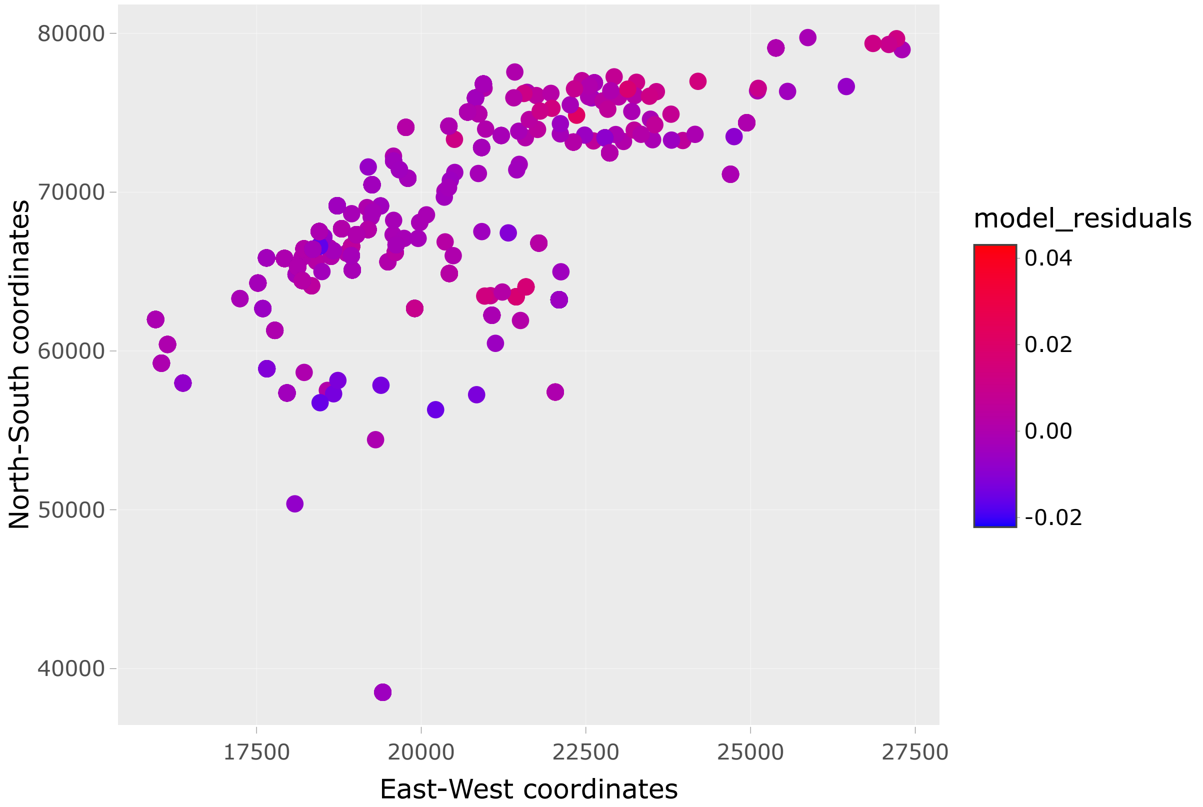
In the left plot we can see a plot of the residuals vs the fitted values. If we take a cluser look on this plot we can notice a funnel shape(as the fitted values goes higher the variance of the residuals gets higher as wel) so there might be heteroskedasticity in the model. Furthermore, i wanted to examine how does the ratio between vaccination first dose to population effects the error rates in the model.

In the right plot we can see the relations between the ratio of vaccination first dose to population size and the residuals. The main interasting thing in this plot is that as the ratio of accination first dose to population grows the residuals tend to be cluser to zero.

Also, i wanted to explore the residuals in each city in the data.

```
model_residuals <-res$new_cases
gol<-data %>% ggplot(aes(x=town_east_coord,y=town_north_coord))+geom_point(aes(color=model_residuals,t
ext=town_eng.y),size=2) + scale_colour_gradient(low = "blue", high = "red") + ggtitle("Model Residuals
In Cities") + xlab("East-West coordinates") + ylab("North-South coordinates")
ggplotly(gol,tooltip = c("text"))
```

Model Residuals In Cities



In the plot we can see each city in it's (x,y) location with the given coordinates and a colour gradiant that marks the model residual in the city. Form this figure we can see that the type of population in the city may effect the model predictions. The residuals in ethnic minority cities in the north and south have higher and lower residual values which means it is hader for the model to get right predictions in thouse cities then other cities in the figure. Also can be seen in the figure that the model is more effective in center Isreal than the periphery areas in Israel.

d.

```
cali <- cbind(for_cal,data$agas_code,data$town_eng.y)
later <-cali %>% select(`data$agas_code`,`data$town_eng.y`)
cali <- cali %>% select(-`data$town_eng.y`,-`data$agas_code`)
weight<-function(X){
  return(solve(t(X)%*%X + diag(model1$model_lambda,dim(t(X)%*%X)[2]))%*%t(X))
}
W<-weight(as.matrix(cali))
my_chose <- later[2,] # first non zero agas is Ofaqim ,agas=1
tt <- as.matrix(cali[2,])%*%as.matrix(W)
tos <- cbind(t(tt),later)
ord <- order(tos$`2`,decreasing = T)
weight_data <- cbind(cali,tos)
top5_with_agas <- weight_data[c(2,ord[1:5]),]
row.names(top5_with_agas)[1:6] <- c("Area chosen","1","2","3","4","5")
top5_with_agas <- top5_with_agas %>% select(-`1`,-`data$agas_code`)
colnames(top5_with_agas)[c(1:dim(top5_with_agas)[2])]<- c("% first dose","% in clalit","town income",
"% bagrut","town socioeconimic index","% pop over 20","%  pop over 50","% pop over 70","Weights","cit
y")
top5_with_agas[,1:9]<- round(top5_with_agas[,1:9],3)
wee <-top5_with_agas[2:6,9:10]
kable(t(wee))
```

	1	2	3	4	5
Weights	0.009	0.007	0.007	0.007	0.007
city	KAFAR KAMA	MAZRA'A	SHIBLI-UMM AL-GHANAM	OFAQIM	JERUSALEM

```
kable(top5_with_agas[,-9])
```

	% first dose	% in clalit	town income	% bagrut	town socioeconimic index	% pop over 20	% pop over 50	% pop over 70	city
Area chosen	18.204	66.4	7123.442	45.195	-0.698	69.787	29.928	9.360	OFAQIM
1	21.524	96.7	8428.656	64.286	0.099	70.738	27.750	6.937	KAFAR KAMA
2	6.872	93.9	6515.066	64.706	-0.422	68.567	23.823	4.149	MAZRA’A
3	11.151	96.2	6563.923	53.103	-0.832	57.591	17.644	3.288	SHIBLI-UMM AL-GHANAM
4	25.352	66.4	7123.442	45.195	-0.698	75.189	36.062	13.812	OFAQIM
5	13.830	41.9	7529.436	37.537	-0.919	77.575	19.180	10.046	JERUSALEM

Answer
There are some similarities between between the rows with the “heavier” weights to the choosen observation. In the table we can see that the first 2 observations with biggest weights have the similar population distribution to the chosen observation (about 70% over 20,23%-29% over 50 and 4%-9% over 70).Also, we can see that the main difference between the first and second observation is in % first dose which is cluser in the first observation then the second. The other observations are have different population distribution then the chosen observation but are similer in other aspects like socioeconomic index and % bagrut but it seems that the model gives more importance to the population distribution then other variables.

3. The model i chose to predict result variable is Elastic Net. This model lets me use combination of penalties(if alpha is clouser to 0 we will have a ridge penalty and if alpha is clouser to 1 we will have a lasso penalty) So, in order to build the model i will iterate over values of alpha and choose the alpha(between 0 and 1) that minimize the mse. Also, because of the structure of the data i will use LOOCV to avoid cross-validation stat split data unequally (in terms having sub sets with under-representation of cities)

```
set.seed(37)
 all_data <- cbind(x,y) # bind the data
  agas_code0_train <- all_data %>% filter(agas_code==0) %>%  sample_frac(0.7) # sample from the small
 cities

  agas_code0_test <- anti_join(all_data %>% filter(agas_code==0),agas_code0_train,by="town_code") # cr
eate test set for agas # code =0

  agas_code_pos_train <- all_data %>% filter(agas_code!=0) %>% group_by(town_code) %>%  sample_frac(0.
7) # sample from the large cities such  #that from each city we will sample number agas proportional t
o the city.
  agas_code_pos_test <- anti_join(all_data %>%filter(agas_code!=0),agas_code_pos_train,by=c("town_cod
e","agas_code"))
  # create test set for agas code !=0

  train_x <-as.data.frame(rbind(as.data.frame(agas_code0_train),as.data.frame(agas_code_pos_train))) #
join the two agas code train sets to one
  test_x <- as.data.frame(rbind(as.data.frame(agas_code0_test),as.data.frame(agas_code_pos_test))) # j
oin the two agas code test sets to one

train_y <- train_x[,dim(train_x)[2]] # split the data to x and y
test_y <- test_x[,dim(test_x)[2]]
train_x <- train_x[,-dim(train_x)[2]]  %>% select(-town_code,-agas_code)
test_x <- test_x[,-dim(test_x)[2]]  %>% select(-town_code,-agas_code)
```

```r
# create table to fit loocv results
tuning_grid = tibble::tibble(
alpha = seq(0, 1, by=0.1),mse_min = NA,mse_1se = NA,lambda_min = NA,
lambda_1se = NA
)
# iterate across alpha values to find the best one
for (i in seq_along(tuning_grid$alpha)){
# fit LOOCV model for each alpha value
fit <- cv.glmnet(as.matrix(train_x),as.matrix(train_y),
alpha=tuning_grid$alpha[i],nfolds = dim(train_x)[1],grouped = FALSE)
# extract MSE and lambda values
tuning_grid$mse_min[i] <- fit$cvm[fit$lambda == fit$lambda.min]
tuning_grid$mse_1se[i] <-fit$cvm[fit$lambda == fit$lambda.1se]
tuning_grid$lambda_min[i] <- fit$lambda.min
tuning_grid$lambda_1se[i] <- fit$lambda.1se
}
kable(tuning_grid)
```

alpha	mse_min	mse_1se	lambda_min	lambda_1se
0.0	0.0000430	0.0000474	0.0005878	0.0087281
0.1	0.0000427	0.0000472	0.0000660	0.0057425
0.2	0.0000427	0.0000469	0.0000763	0.0041657
0.3	0.0000427	0.0000469	0.0000672	0.0033451
0.4	0.0000427	0.0000468	0.0000553	0.0027534
0.5	0.0000427	0.0000469	0.0000486	0.0024175
0.6	0.0000427	0.0000470	0.0000405	0.0022110
0.7	0.0000427	0.0000467	0.0000381	0.0018951
0.8	0.0000427	0.0000470	0.0000333	0.0018199
0.9	0.0000427	0.0000468	0.0000296	0.0016177
1.0	0.0000427	0.0000466	0.0000267	0.0014559

It seems that lasso penalty get the lowest mse and lowest mse se so i will use lasso penalty for my model with the min lambda.

```r
final_model <- glmnet(as.matrix(train_x),as.matrix(train_y),alpha = 1,lambda =0.0000267) # implement lasso regression with minimum lambda
predicti <- predict(final_model, s=final_model$lambda,as.matrix(test_x))
predict_rmse<-sqrt(mean((test_y - predicti)^2)) # calculate RMSE
# organize the test data to the model
test_data <- test_data %>% select(-town,-mahoz,-town_eng.y,-town_pop_denisty,-town_diabetes_rate,-agas_socioeconomic_index,-town_north_coord,-town_east_coord,-X,-town_code,-agas_code)
test_data$accumulated_vaccination_first_dose <-100*test_data$accumulated_vaccination_first_dose/test_data$population
test_data$pop_over20 <-100*test_data$pop_over20/test_data$population
test_data$pop_over50 <-100*test_data$pop_over50/test_data$population
test_data$pop_over70 <-100*test_data$pop_over70/test_data$population
test_data <- cbind(1,test_data)
test_data <- test_data %>% select(-population)
# create predictions
predict_y <- predict(final_model, s=final_model$lambda,as.matrix(test_data))
save(predict_y, predict_rmse, file = "206094278.rda")
```