Graduate School of Business
Stanford University
Fall 2024

OIT249: Data & Decisions
[D&D]

Final Examination – Fall 2024

PLEASE READ THESE INSTRUCTIONS CAREFULLY:

- This exam is available from 9:00AM to 5:00PM on Wednesday, December 11.
- You must upload your solutions to this take-home exam through Canvas, within 4 hours after you downloaded the exam, and no later than 5:00PM on Wednesday, December 11. This means that, in order to have the full 4 hours, you must start the exam before 1:00PM. Exams submitted after 5:00PM will not be graded and will receive a grade of zero.
- You are only allowed one submission.
- Due to limitations on the Canvas platform, for every file upload question, you must upload a single file (pdf) or an archive file (e.g., .zip, .rar), containing all your work. Our preference is that you upload a single pdf file.
- This is an open-notes, open-book, and open-internet examination. You may use any computer to complete the exam. You may ask questions to a Large Language Model (LLM), but ultimately, you are responsible for your answers.
- This examination is intended to be an **individual** effort, and your solutions must reflect only your work. You *may not* consult with or accept help from anyone during this exam. This provision applies to all phases of the exam: reading the questions, preparing your solution, checking your answers, and preparing your submission. All provisions of the Stanford Honor Code apply. By taking this exam, you are agreeing to the following:

  > In recognition of and in the spirit of the Stanford Honor Code, I certify that I will neither receive nor give unpermitted aid on this examination, and that I will report, to the best of my ability, all Honor Code violations that I observe.

- The examination consists of 4 problems, each with several subproblems. The total number of points on the exam is 100. Each problem states the number of points it is worth. Allocate your time accordingly.
- Show your work. Unless otherwise indicated, partial credit may be given for partially correct work. Carry out all additions, subtractions, and multiplications. You may show your work by specifying the Excel or R command associated with an answer (e.g. NORMDIST(1.28,0,1,1) or (Data$Age)). Final answers may also be left in fractional form.

- If you feel that a problem could be interpreted in several ways or you are unsure about what a question is asking, state your interpretation and assumptions clearly, and proceed accordingly. We will take your explanation into account when grading.
- If you have questions during the exam, you can feel free to e-mail me at `soma@stanford.edu`, or Vishal at `jainvi@stanford.edu` or Karina at `ksantoso@stanford.edu`, and we will try our best to respond promptly. There may be times when one or all of us are unavailable. In that case, please see the bullet point above.
- If you experience a Canvas technology issue while trying to upload your exam, please **immediately** e-mail us and attach a copy of your exam answers.
- Use R script files and frequently save your work to disk, as insurance against a computer problem.
- ***Good Luck.***

Question 1. (20 points)   Excess Deaths due to Covid-19

This is an analysis of excess deaths due to Covid-19. It is based on weakly data by country collected from The New York Times and The Economist from December 2019 to July 2022. The key variables in the dataset are: country and excess_deaths. There are 41 countries. Excess deaths are computed as the differences between total deaths observed (total_deaths) and expected deaths given the demographic characteristics of the country (expected_deaths). For this exercise, you will not be provided any data. You will be able to answer the questions with the information on the following R output:

```r
table(covid$country)
```

```
[commandchars=\\\{\}]
##
##      Australia        Austria        Belgium        Britain       Bulgaria
##            117            131            129            131            127
##         Canada          Chile       Colombia        Croatia         Cyprus
##            113            132            132            127            127
## Czech Republic        Denmark        Ecuador        Estonia        Finland
##            126            132            133            131            127
##         France        Germany         Greece      Guatemala        Hungary
##            128            131            127            104            127
##        Iceland           Iran         Israel          Italy         Latvia
##            127            132            131            122            130
##      Lithuania     Luxembourg          Malta         Mexico     Montenegro
##            128            127            128            124             96
##    Netherlands    New Zealand         Norway           Peru         Poland
##            132            131            132            132            130
##       Portugal        Romania       Slovakia       Slovenia   South Africa
##            131            126            124            129            134
##    South Korea          Spain         Sweden    Switzerland        Tunisia
##            122            126            128            130             58
##  United States
##            128
```

```r
reg1 = lm(excess_deaths_per_100k ~ factor(country), data = covid)
summary(reg1)
```

```
##
## Call:
## lm(formula = excess_deaths_per_100k ~ factor(country), data = covid)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.5958  -2.0319  -0.5695   1.0458  30.7688
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     0.18758    0.35839   0.523 0.600723
## factor(country)Austria          1.51061    0.49311   3.063 0.002198 **
## factor(country)Belgium          1.52402    0.49491   3.079 0.002084 **
## factor(country)Britain          1.61626    0.49311   3.278 0.001053 **
## factor(country)Bulgaria         7.84468    0.49676  15.792  < 2e-16 ***
## factor(country)Canada           0.51810    0.51130   1.013 0.310957
## factor(country)Chile            1.82435    0.49223   3.706 0.000212 ***
## factor(country)Colombia         2.48326    0.49223   5.045 4.68e-07 ***
## factor(country)Croatia         4.07396    0.49676   8.201 2.92e-16 ***
## factor(country)Cyprus          1.08728    0.49676   2.189 0.028655 *
## factor(country)Czech Republic  3.18564    0.49770   6.401 1.67e-10 ***
## factor(country)Denmark         0.30539    0.49223   0.620 0.534997
## factor(country)Ecuador         2.96706    0.49136   6.039 1.65e-09 ***
## factor(country)Estonia         2.43818    0.49311   4.945 7.85e-07 ***
## factor(country)Finland         0.84824    0.49676   1.708 0.087776 .
## factor(country)France          1.06338    0.49583   2.145 0.032023 *
## factor(country)Germany         0.96202    0.49311   1.951 0.051115 .
## factor(country)Greece          2.29006    0.49676   4.610 4.11e-06 ***
## factor(country)Guatemala       2.16544    0.52243   4.145 3.45e-05 ***
## factor(country)Hungary         3.32416    0.49676   6.692 2.42e-11 ***
## factor(country)Iceland         0.18501    0.49676   0.372 0.709579
## factor(country)Iran            2.15607    0.49223   4.380 1.21e-05 ***
## factor(country)Israel          0.74960    0.49311   1.520 0.128529
## factor(country)Italy           2.62459    0.50162   5.232 1.73e-07 ***
## factor(country)Latvia          3.64801    0.49400   7.385 1.75e-13 ***
## factor(country)Lithuania       5.66288    0.49583  11.421  < 2e-16 ***
## factor(country)Luxembourg      0.07708    0.49676   0.155 0.876694
## factor(country)Malta           0.78882    0.49583   1.591 0.111684
## factor(country)Mexico          4.00207    0.49963   8.010 1.38e-15 ***
## factor(country)Montenegro      4.63144    0.53383   8.676  < 2e-16 ***
## factor(country)Netherlands     1.38944    0.49223   2.823 0.004778 **
## factor(country)New Zealand    -0.35938    0.49311  -0.729 0.466150
## factor(country)Norway          0.37025    0.49223   0.752 0.451971
## factor(country)Peru            4.18447    0.49223   8.501  < 2e-16 ***
## factor(country)Poland          3.46253    0.49400   7.009 2.67e-12 ***
## factor(country)Portugal        1.93031    0.49311   3.915 9.16e-05 ***
## factor(country)Romania         4.94743    0.49770   9.941  < 2e-16 ***
## factor(country)Slovakia        4.00784    0.49963   8.022 1.26e-15 ***
## factor(country)Slovenia        1.94131    0.49491   3.923 8.86e-05 ***
## factor(country)South Africa    3.20991    0.49050   6.544 6.50e-11 ***
## factor(country)South Korea     0.40287    0.50162   0.803 0.421929
## factor(country)Spain           1.77713    0.49770   3.571 0.000359 ***
## factor(country)Sweden          0.75697    0.49583   1.527 0.126897
## factor(country)Switzerland     1.12388    0.49400   2.275 0.022940 *
## factor(country)Tunisia         0.41368    0.62253   0.665 0.506384
## factor(country)United States      XXXX    0.49583   5.045     XXXX
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.877 on 5724 degrees of freedom
## Multiple R-squared:  0.1588, Adjusted R-squared:  0.1522
## F-statistic: 24.01 on 45 and 5724 DF,  p-value: < 2.2e-16
```

Based on the previous output, answer the questions below:

(a) (3 points) What is the coefficient estimate on the US dummy variable?

(b) (3 points) Interpret the coefficient on the US dummy variable.

(c) (3 points) What is the p-value of the coefficient on the US dummy variable?

(d) (3 points) What is the hypothesis test that the p-value correspond to?

(e) (3 points) Were the excess deaths in the U.S particularly different than the reference group?

We create a new variable called lagged_excess_deaths that is the lagged value of excess_deaths_per_100k (i.e., the excess deaths per 100k corresponding to the previous week). We then run a regression with lagged_excess_deaths and country dummies.

```
covid_clean <- covid[!is.na(covid$lagged_excess_deaths), ]
panel_model_man <- lm(excess_deaths_per_100k ~ lagged_excess_deaths + factor(country), data = covid_clea
summary(panel_model_man)
```

```
## 
## Call:
## lm(formula = excess_deaths_per_100k ~ lagged_excess_deaths + 
##     factor(country), data = covid_clean)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -10.3561  -0.7467  -0.1118   0.6450  20.2819 
## 
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                    0.027571   0.157133   0.175 0.860722    
## lagged_excess_deaths           0.896960   0.005785 155.041  < 2e-16 ***
## factor(country)Austria         0.179282   0.216327   0.829 0.407277    
## factor(country)Belgium         0.171661   0.217125   0.791 0.429205    
## factor(country)Britain         0.171206   0.216353   0.791 0.428786    
## factor(country)Bulgaria        0.841780   0.222533   3.783 0.000157 ***
## factor(country)Canada          0.047994   0.224212   0.214 0.830510    
## factor(country)Chile           0.188925   0.216021   0.875 0.381847    
## factor(country)Colombia        0.260439   0.216244   1.204 0.228496    
## factor(country)Croatia         0.432900   0.219062   1.976 0.048186 *  
## factor(country)Cyprus          0.069265   0.217860   0.318 0.750548    
## factor(country)Czech Republic  0.314242   0.218980   1.435 0.151334    
## factor(country)Denmark         0.030705   0.215768   0.142 0.886843    
## factor(country)Ecuador         0.309951   0.216070   1.434 0.151487    
## factor(country)Estonia         0.304785   0.216605   1.407 0.159452    
## factor(country)Finland         0.096677   0.217819   0.444 0.657173    
## factor(country)France          0.102278   0.217442   0.470 0.638109    
## factor(country)Germany         0.106349   0.216222   0.492 0.622843    
## factor(country)Greece          0.258475   0.218164   1.185 0.236158    
## factor(country)Guatemala       0.220481   0.229468   0.961 0.336677    
## factor(country)Hungary         0.353976   0.218626   1.619 0.105483    
## factor(country)Iceland        -0.013595   0.217765  -0.062 0.950222    
## factor(country)Iran            0.217344   0.216129   1.006 0.314642    
## factor(country)Israel          0.071751   0.216194   0.332 0.739992    
## factor(country)Italy           0.275321   0.220448   1.249 0.211747    
## factor(country)Latvia          0.384313   0.217582   1.766 0.077400 .  
## factor(country)Lithuania       0.606246   0.219824   2.758 0.005837 ** 
## factor(country)Luxembourg      0.051800   0.217762   0.238 0.811988    
## factor(country)Malta           0.099253   0.217398   0.457 0.648013    
## factor(country)Mexico          0.398516   0.220282   1.809 0.070485 .  
## factor(country)Montenegro      0.671707   0.235662   2.850 0.004384 ** 
## factor(country)Netherlands     0.149882   0.215913   0.694 0.487600    
## factor(country)New Zealand    -0.040439   0.216160  -0.187 0.851604    
## factor(country)Norway          0.046147   0.215771   0.214 0.830654    
## factor(country)Peru            0.424081   0.217143   1.953 0.050868 .  
## factor(country)Poland          0.346312   0.217493   1.592 0.111376    
## factor(country)Portugal        0.226238   0.216440   1.045 0.295943
## factor(country)Romania         0.509481   0.220096   2.315 0.020659 *
## factor(country)Slovakia        0.413673   0.220276   1.878 0.060436 .
## factor(country)Slovenia        0.200212   0.217243   0.922 0.356774
## factor(country)South Africa    0.328836   0.215810   1.524 0.127632
## factor(country)South Korea     0.048578   0.219920   0.221 0.825186
## factor(country)Spain           0.200437   0.218424   0.918 0.358842
## factor(country)Sweden          0.078376   0.217396   0.361 0.718469
## factor(country)Switzerland     0.131689   0.216643   0.608 0.543304
## factor(country)Tunisia         0.033709   0.273756   0.123 0.902006
## factor(country)United States   0.262311   0.217839   1.204 0.228583
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.692 on 5677 degrees of freedom
## Multiple R-squared:  0.8399, Adjusted R-squared:  0.8386
## F-statistic: 647.6 on 46 and 5677 DF,  p-value: < 2.2e-16
```

(f) (3 points) Based on this regression, were the excess deaths in the U.S particularly different than the reference group?

(g) (2 points) Explain why you arrived to the same (or different) conclusion in questions (e) and (f).

Question 2. (20 points)   Data Scientist Salaries

This exercise involves analyzing a dataset of data science salaries. The dataset, `ds_salaries.csv`, contains information about employees in the data science field. Your task is to explore the data and build regression models to understand how different variables influence salaries. The dataset includes the following variables:

- `work_year`: The year the salary data was recorded.
- `experience_level`: The experience level of the employee (e.g., EN = Entry-level, MI = Mid-level, SE = Senior, EX = Executive).
- `employment_type`: The type of employment (e.g., FT = Full-time, PT = Part-time, CT = Contract, FL = Freelance).
- `job_title`: The title of the employee's job.
- `salary`: The salary reported in the local currency.
- `salary_currency`: The currency in which the salary is reported.
- `salary_in_usd`: The salary converted to USD.
- `employee_residence`: The country where the employee resides.
- `remote_ratio`: The percentage of the job that can be done remotely.
- `company_location`: The country where the company is located.
- `company_size`: The size of the company (e.g., S = Small, M = Medium, L = Large).

(a) (5 points) Build a linear regression model to predict `salary_in_usd` using `remote_ratio` and the categorical variables: `experience_level`, `employment_type`, `company_size`, and `work_year`.

Interpret the coefficients for:

- `employment_type` (equal to Full Time).
- `remote_ratio`.

(b) (5 points) Plot the residuals versus the predicted salary. Is the mean of the residuals zero for every value of predicted salary? Is the spread (variance) constant for every value of predicted salary?

(c) (5 points) Build another linear regression model using the same predictors, but with the log of `salary_in_usd` as the dependent variable. Compare this model to the previous one. How does the interpretation of the coefficients for `employment_type` (equal to Full Time) and `remote_ratio` change?

(d) (5 points) For the log-transformed model, plot the residuals versus the predicted salary. Is the mean of the residuals zero for every value of predicted salary? Is the spread (variance) constant for every value of predicted salary?

Question 3. (30 points) Condo Prices and Characteristics

This exercise involves analyzing a dataset of condos listed for sale. The dataset, `redfin_condos.csv`, contains information about condos, including their prices and various characteristics. Your task is to explore the data and build regression models to understand how these characteristics influence condo prices. The dataset includes the following variables:

- `PRICE`: The listed price of the condo (in dollars).
- `SQUARE.FEET`: The total square footage of the condo.
- `BEDS`: The number of bedrooms in the condo.
- `BATHS`: The number of bathrooms in the condo.
- `LOT.SIZE`: The lot size of the condo property (in square feet).
- `HOA.MONTH`: Monthly Home Owner Association fee.
- `LATITUDE`: The latitude of the condo's location.
- `LONGITUDE`: The longitude of the condo's location.

(a) (5 points) Build a linear regression model to predict `PRICE` using the variables `SQUARE.FEET`, `LOT.SIZE`, `HOA.MONTH`,`BEDS`, and `BATHS`.

Interpret the coefficients for:

- `SQUARE.FEET`.
- `BEDS`.

(b) (5 points) Plot the residuals versus the predicted prices from the model in part 1.

- Is the mean of the residuals zero for every value of predicted price?
- Is the spread (variance) constant for every value of predicted price?

(c) (5 points) Modify the regression model from part 1 by including an interaction term between `SQUARE.FEET` and `LATITUDE`, i.e., add the product of `SQUARE.FEET` and `LATITUDE` as an additional variable in the regression. How does the inclusion of this interaction affects the interpretation of the coefficient of `SQUARE.FEET`?

(d) (5 points) Build a new linear regression model to predict the log-transformed `PRICE` (i.e., `log(PRICE)`) using `log(SQUARE.FEET)`, `log(LOT.SIZE)`, `log(HOA.MONTH)`, `BEDS`, and `BATHS`.

Interpret the coefficients for:

- `log(SQUARE.FEET)`

- `BEDS`

(e) (5 points) Using the results in this regression, test the hypothesis that, keeping all other variables constant, the price of a condo in Dallas is proportional to its square footage, e.g., a house twice as big sells for twice as much.

(f) (5 points) Plot the residuals versus the predicted log-transformed prices from the model in part (d).

- Is the mean of the residuals zero for every value of predicted log price?
- Is the spread (variance) constant for every value of predicted log price?

Question 4. (30 points) **Predicting Loan Repayment**

This exercise involves building a predictive model for loan repayment using data from LendingClub, a former peer-to-peer lender. There are two datasets:

1. **loans_data.csv**: This dataset will be used to train the model. It contains the dependent variable `fully_paid` (indicating whether a loan was fully paid) and various independent variables you can use as predictors.

2. **loans_to_predict.csv**: This dataset contains the loans for which you will predict the probability of repayment. It includes the same variables as the training dataset, except for the dependent variable `fully_paid`.

The variables included in these files are:

Variable	**Description**
X	Unique identifier for each loan record.
loan_amnt	The total amount of the loan applied for by the borrower.
funded_amnt	The amount of the loan that was actually funded.
funded_amnt_inv	The amount of the loan funded by investors.
term	Loan repayment term in months (e.g., 36 or 60 months).
int_rate	The interest rate of the loan.
installment	The monthly payment owed by the borrower.
grade	LendingClub-assigned loan grade (e.g., A, B, C).
sub_grade	More granular sub-grade of the loan (e.g., A1, A2, B4).
emp_title	Job title of the borrower.
emp_length	Length of employment in years.
home_ownership	Homeownership status of the borrower (e.g., RENT, OWN, MORTGAGE).
annual_inc	Borrower's self-reported annual income.
verification_status	Status of income verification (e.g., "not verified," "source verified").
issue_d	The month and year the loan was issued.
pymnt_plan	Payment plan indicator (e.g., "n" for no).
desc	Loan description provided by the borrower (optional).
purpose	Purpose of the loan (e.g., "debt_consolidation," "small_business").
title	Loan title or category as provided by the borrower.
zip_code	First three digits of the borrower's zip code.
addr_state	State where the borrower resides.
dti	Debt-to-income ratio of the borrower.
delinq_2yrs	Number of delinquent accounts in the past 2 years.
earliest_cr_line	Date of the borrower's earliest reported credit line.
fico_range_low	Lower bound of the borrower's FICO score range.

fico_range_high	Upper bound of the borrower’s FICO score range.
inq_last_6mths	Number of credit inquiries in the past 6 months.
mths_since_last_delinq	Months since the borrower’s last delinquency. (NA if no delinquency).
mths_since_last_record	Months since the borrower’s last derogatory public record. (NA if no record).
open_acc	Number of open credit lines in the borrower’s credit file.
pub_rec	Number of derogatory public records.
revol_bal	Total credit revolving balance.
revol_util	Revolving line utilization rate (as a percentage).
total_acc	Total number of credit accounts.
initial_list_status	If a loan was initially listed in the whole (w) or fractional (f) market.
fully_paid	Status of the loan: 1 if fully paid and 0 if charged off or late (only included in `loans_data.csv`.

1. (10 points) Train a predictive model:

   - Use the data in `loans_data.csv` to build a model predicting the event that a loan is fully paid (`fully_paid`) based on the other variables in the dataset. You may also create new variables. You may try several of the methods we learned in class and pick the one you think will work best to predict the loans in `loans_to_predict.csv`.

2. (5 points) Make predictions:

   - Use your trained model to predict the probability that each loan in `loans_to_predict.csv` will be fully paid.
   - Add a new column to the dataset in `loans_to_predict.csv` called `pred` containing these predicted probabilities.

3. (5 points) Output:

   - Create a new CSV file with the loans in `loans_to_predict.csv` containing only the following two columns:
     - `X`: The unique loan identifier.
     - `pred`: The column with the predicted probabilities.

     Hint: to keep only these two variables you can use the command:

     `write.csv(loans_to_predict[,c(’X’,’pred’)], ’yourname.csv’)`

     This will create a file called `yourname.csv` with only the columns `X` and `pred`.

4. Submit your code used for the analysis (R, or other software) using the “Choose a File” button under “Question 4”. Your grade of parts 1-3 will be based on this code.

5. (10 points) Submit the resulting CSV file with the predictions using the "Choose a File" button under "Question 5". Your grade of part 4 will depend on the RMSE of your `pred` column relative to the actual repayment of the loans in `loans_to_predict.csv` which were not given to you.