BDA Project

Niko Miller, Akseli Manninen and Santeri Löppönen

Contents

1. Project Description and Motivation

Studying the relationship between income and education has been the focus on many studies. The studies have concluded that a strong connection exists between higher education and income (Card, 1999). In general, individuals with stronger education are more likely to be employed and earn a big salary compared to less educated people (Card, 1999). For that reason, education is described as an investment in human capital (Wolla & Sullivan, 2017).

This study examines this phenomenon from the perspective of people that have acquired their education from colleges in the United States. As the connection between education and income has been shown in the existing literature, this study strives to further examine the associations between college related features and income level years after graduation. This project is not limited to only considering educational aspects but expands it to family backgrounds.

In this study, a Bayesian approach is taken to observe the bond between the educational and family related features and earnings. It is in our interest to find out how accurately the selected features can predict future income for the students. Furthermore, finding well-predictive features among the vast number of variables is pursed, and then evaluating predictive performance using these features with a selected statistical models.

As this study is conducted in a university environment by university students and presented mainly to other students and faculty, the findings of this study could be especially meaningful for the members of the group and peers on the course.

2. Data and the analysis problem

The used dataset is the most recent institutional-level college scorecard data from the US Department of Education. The institutional-level dataset contains aggregate data for each educational institution and includes data on institutional characteristics, enrollment, student aid, costs and student outcomes. The dataset has over 6000 observations on more than 3000 variables.

This dataset was chosen because it seemed to provide information that could answer interesting education-related questions in the project, and also because the dataset has a large number of observations and a large number of variables. The large amount of variables permits for a lot of flexibility when it comes to modelling with the data and also having enough data is important in order to be able to make valid inferences.

After investigating the dataset, the most intriguing analysis problem was to study how college related factors affect the median earnings 10 years after entry.

```r
# read in data sets
data <- read.csv2("./Data/Most-Recent-Cohorts-Institution.csv",
                  sep = ",", fileEncoding="UTF-8-BOM") %>% as_tibble()
data.description <- read.csv2("./Data/CollegeScorecardDataDictionary.csv",
                              sep = ",", fileEncoding="UTF-8-BOM") %>% as_tibble()
```

Feature selection

Phase 1 - Feature selection based on literature and domain knowledge

The initial data set had more than 3000 features and in that regards, the number of observations is relatively small (a relatively wide dataset). There are also a lot of missing values in the data set and some features are missing. For these reasons, there was a need to prune features.

The used process of feature selection consisted of two phases: In the first phase, *a subset of features was select based on the features used in the existing literature* - LÄHDE? and using domain knowledge. From the potential features, only those having enough data were included in the subset and others were discarded.

The selected independent variables in the first phase were

	Name	Data type	Description
1	SATVRMID	integer	Midpoint of SAT scores at the institution (critical reading)
2	SATMTMID	integer	Midpoint of SAT scores at the institution (math)
3	SATWRMID	integer	Midpoint of SAT scores at the institution (writing)
4	MD_FAMINC	double	Median family income
5	AGE_ENTRY	double	Average age of entry
6	FEMALE	double	Share of female students
7	FIRST_GEN	double	Share of first-generation students
8	PCT_WHITE	double	Percent of the population from students' zip codes that is White
9	DEBT_MDN_SUPP	integer	Median debt, suppressed for n=30
10	C150_4	double	Completion rate for first-time, full-tim students
11	COSTT4_A	integer	Average cost of attendance (academic year institutions)
12	POVERTY_RATE	double	Poverty rate
13	UNEMP_RATE	double	Unemployment rate
14	MARRIED	double	Share of married students
15	VETERAN	double	Share of veteran students
16	LOCALE	categorical	Locale of institution
17	CCBASIC	categorical	Carnegie Classification – basic
18	CONTROL	categorical	Control of institution

Dependent variable:

Name	Data type	Description
MD_EARN_WNE_P10	double	Median earnings of students 10 years after entry

Code for extracting the feature subset

```r
# DATA MANIPULATION ----

# list variables
id.vars <- c("UNITID", "INSTNM", "CITY", "ST_FIPS", "REGION")
numerical.vars <- c("SATVRMID", "SATMTMID", "MD_FAMINC", "AGE_ENTRY", "FEMALE",
                    "FIRST_GEN", "PCT_WHITE", "DEBT_MDN_SUPP", "C150_4", "COSTT4_A",
                    "MD_EARN_WNE_P10", "POVERTY_RATE", "UNEMP_RATE", "MARRIED")
categorical.vars <- c("LOCALE", "CCBASIC", "CONTROL")
SAT.vars <- c("SATVRMID", "SATMTMID")

# filter specific school types
schooltype.filter <- seq(14,23)

# create map for variable descriptions
variable.descriptions <- data.description %>%
  select(VARIABLE.NAME, NAME.OF.DATA.ELEMENT, NOTES) %>%
  filter(VARIABLE.NAME %in% c(id.vars, categorical.vars, numerical.vars))

# extract categorical variables
data.categorical <- data %>%
  select(all_of(id.vars), all_of(categorical.vars)) %>%
  mutate(across(.cols = all_of(categorical.vars), .fns = as.factor))
```

```
data.categorical.dropna <- data.categorical %>%
  drop_na()

# extract numerical variables
data.numerical <- data %>%
  select(all_of(id.vars), all_of(numerical.vars)) %>%
  mutate(across(.cols = c(id.vars[1], numerical.vars), .fns = as.numeric))
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(numerical.vars)‘ instead of ‘numerical.vars‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
data.numerical.dropna <- data.numerical %>%
  drop_na()

# aggregate SAT scores
data.numerical.dropna <- data.numerical.dropna %>%
  mutate(SAT_ALL = data.numerical.dropna %>%
           select(SATVRMID, SATMTMID) %>%
           rowMeans()
         )

data.joined <- data.numerical %>%
  inner_join(data.categorical, by = id.vars) %>%
  filter(CCBASIC %in% schooltype.filter)

data.joined.dropna <- data.numerical.dropna %>%
  inner_join(data.categorical.dropna, by = id.vars) %>%
  filter(CCBASIC %in% schooltype.filter)

categorical.vars.data <- data.joined.dropna %>%
  select(all_of(categorical.vars), MD_EARN_WNE_P10) %>%
  relocate(MD_EARN_WNE_P10, 1)


data.joined <- data.numerical %>%
  inner_join(data.categorical, by = id.vars) %>%
  filter(CCBASIC %in% schooltype.filter)

data.joined.dropna <- data.numerical.dropna %>%
  inner_join(data.categorical.dropna, by = id.vars) %>%
  filter(CCBASIC %in% schooltype.filter)
```

Phase 2 - Feature selection with correlation and visual dependency

In the second phase of feature selection, a subset of features was selected from the 18 variables of the first phase. The correlations between the features were examined as well as their associations to the dependent variable.

Numerical variables

SAT scores were combined as one variable, because they were correlated and viewed as one entity. However, writing SAT scores had too few observations, due to discontinued tracking, so the variable SAT_ALL was formed by summing the math and critical thinking SAT scores. SAT scores were included because the correlation was high with the dependent variable.

For the rest of the numerical variables, if the correlation between a feature and the dependent variable was low and there was no observable dependency between the two in the scatter plot, the feature was excluded. To avoid multicollinearity, we removed independent variables that were highly correlated with other independent variables, especially if they were not clearly correlated with the dependent variable and we couldn't form a believable hypothesis for the mechanism through which that variable affected income after college.

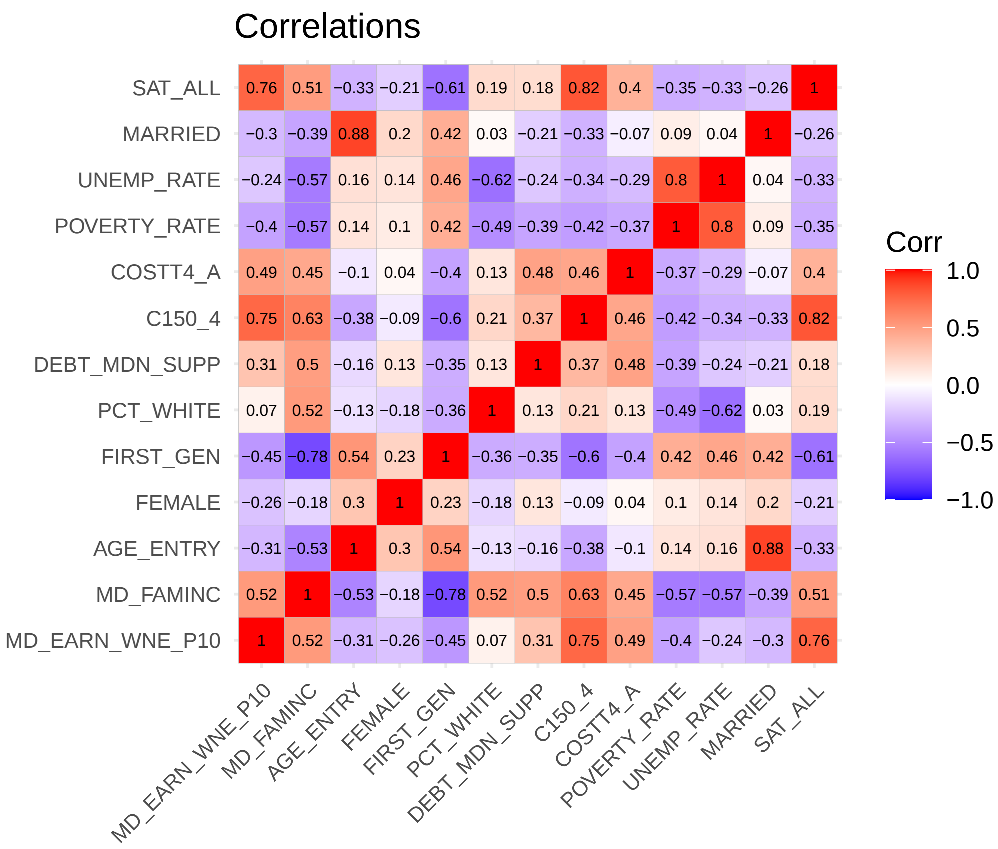
The selected numerical variables were: SAT_ALL (median sum of math and critical thinking SAT scores), MD_FAMINIC (median family income of the student), AGE_ENTRY (median age of starting at the college), COSTT4_A (median cost of college), and POVERTY_RATE (poverty rate in the area the college is located).

Visualizing correlations and data points with the dependent variable.

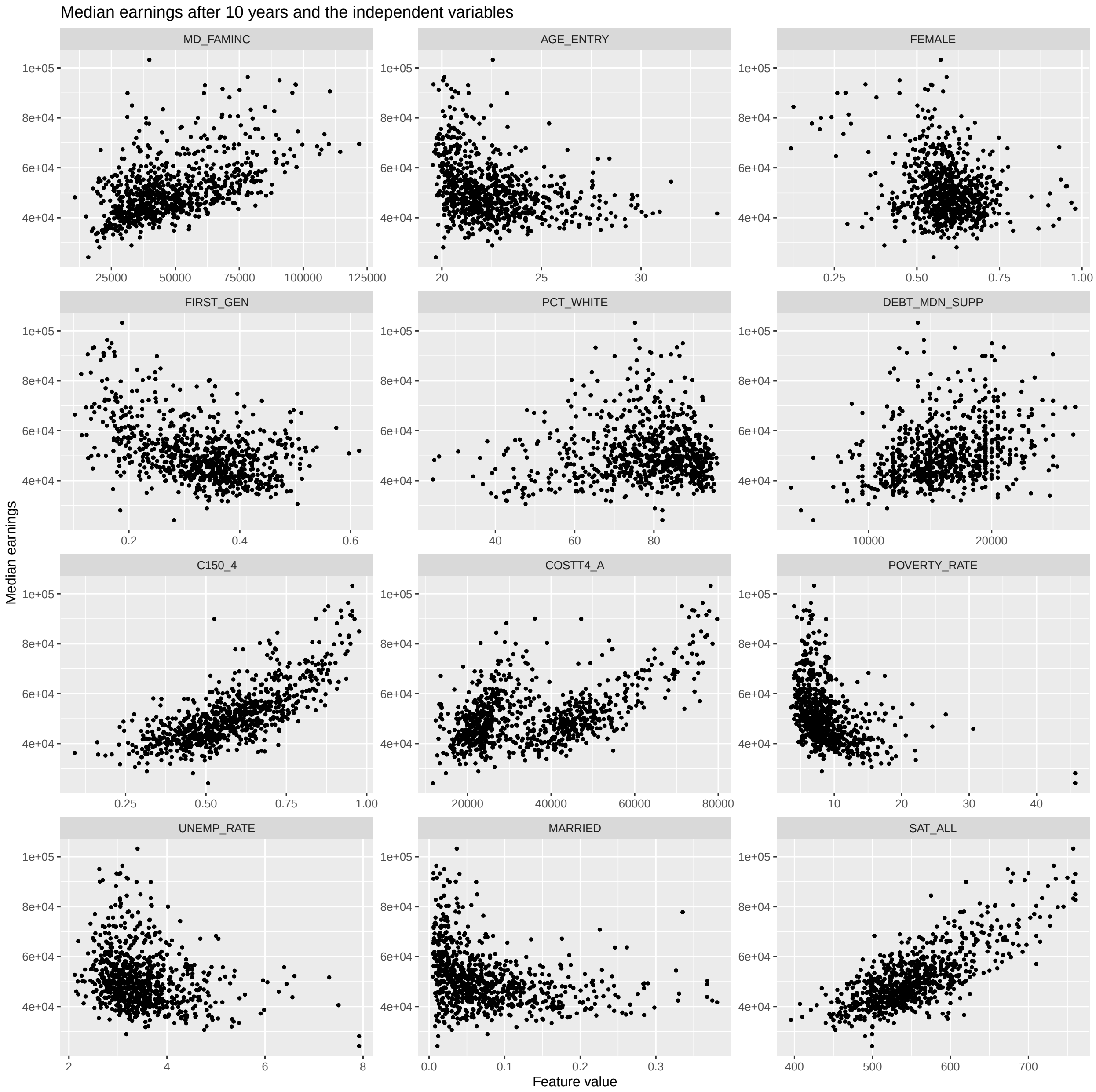
```
# PRELIMINARY ANALYSIS ----

# make variable type specific data frames for plots etc.
numerical.vars.data <- data.joined.dropna %>% select(!c(id.vars,
  categorical.vars, SAT.vars)) %>% relocate(MD_EARN_WNE_P10, 1)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(id.vars)` instead of `id.vars` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(categorical.vars)` instead of `categorical.vars` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
## Note: Using an external vector in selections is ambiguous.
## i Use `all_of(SAT.vars)` instead of `SAT.vars` to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
categorical.vars.data <- data.joined.dropna %>%
  select(all_of(categorical.vars), MD_EARN_WNE_P10) %>%
  relocate(MD_EARN_WNE_P10, 1)

# Correlation plot
ggcorrplot(cor(numerical.vars.data),
          lab = TRUE,
          lab_size = 2,
          title = "Correlations",
          tl.cex = 8,
          )
```



```
melted.numerical.vars.data <- melt(numerical.vars.data, id.vars = "MD_EARN_WNE_P10")

melted.numerical.vars.data <- melt(numerical.vars.data, id.vars = "MD_EARN_WNE_P10")
ggplot(melted.numerical.vars.data, aes(x = value, y = MD_EARN_WNE_P10)) +
  facet_wrap(~variable, scales = "free", ncol = 3) +
  geom_point(shape=20, color="black") +
  ggtitle("Median earnings after 10 years and the independent variables") +
  xlab("Feature value") + ylab("Median earnings")
```



Categorical variables

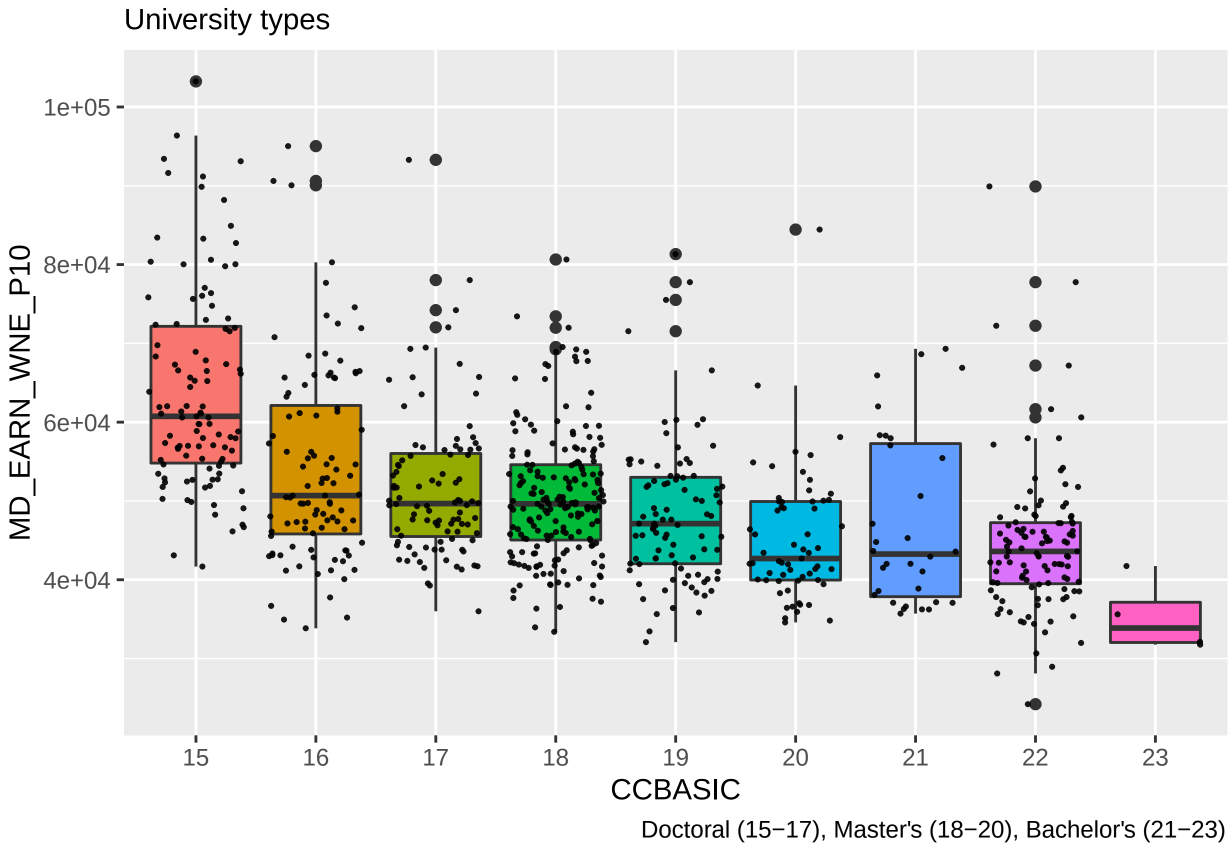
The categorical variables were visualized with box plots to see if income after college differed among the categories. Based on an analysis on the variable LOCALE (whether college located in metropolis, city, suburb, town, or rural area) a new binary variable URBAN was generated, where value 1 represents any area that is not categorized as rural.

The categorical variable CCBASIC which represented the Carnegie Classification was divided into two binary MASTER and DOCTORL. These variables represent if the college is classified as Master's college and university or Doctoral's college and university. If a college does not belong to either of those, it is a Bachelor's college and university. Other special focus colleges and universities were discarded from the dataset to keep the model more simple and avoid unnecessary outliers due to unconventional nature of some very specialized colleges. Our model does not attempt to provide accurate predictions for specialized colleges.

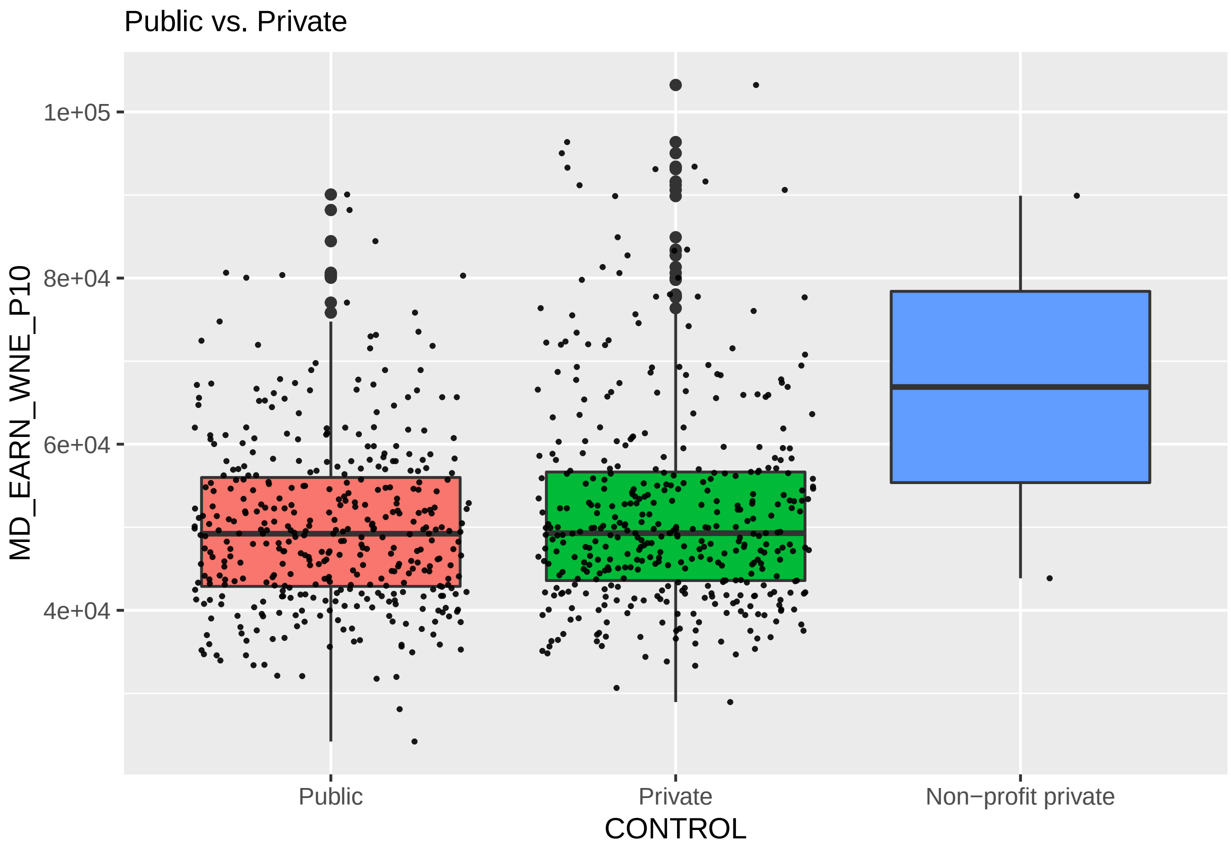
The categorical variable CONTROL had three classes: public, for-profit private and nonprofit private. CONTROL was modified into a binary variable PRIVATE representing if the school is private or public school. There were only few non-profit private observations, so those were merged together with the private observations.

The selected categorical variables were: URBAN, DOCTORAL, MASTER, PRIVATE.

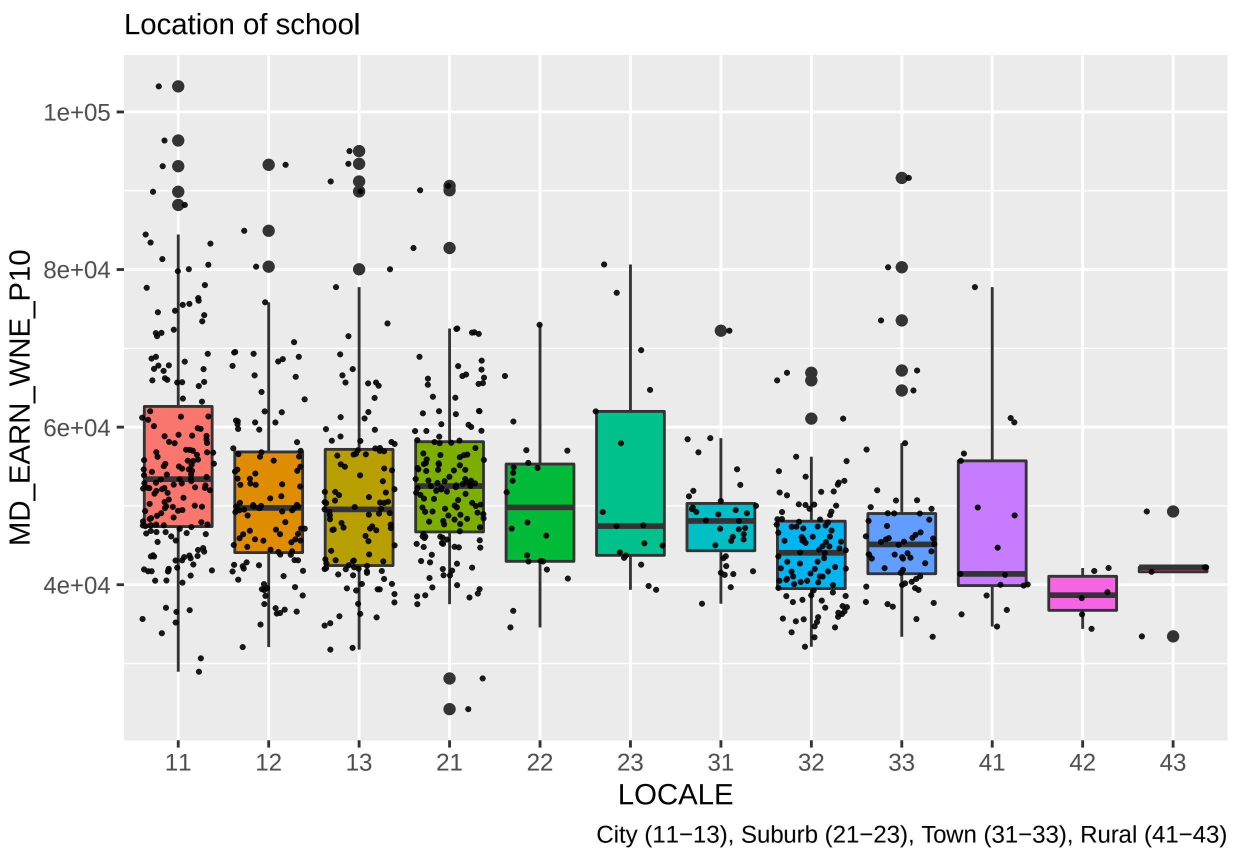
```
# categorical variable box plots
melted.categorial <- melt(categorical.vars.data, id.vars = "MD_EARN_WNE_P10")
melted.categorial.ccbasic <- melted.categorial %>% as_tibble() %>% filter(variable=="CCBASIC")
melted.categorial.control <- melted.categorial %>% as_tibble() %>% filter(variable=="CONTROL")
melted.categorial.locale <- melted.categorial %>% as_tibble() %>% filter(variable=="LOCALE")

ggplot(melted.categorial.ccbasic, aes(x=value, y=MD_EARN_WNE_P10, fill=value)) +
  geom_boxplot() +
  geom_jitter(color="black", size=0.4, alpha=0.9) +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("University types") +
  xlab("CCBASIC") +
  labs(caption = "Doctoral (15-17), Master's (18-20), Bachelor's (21-23)")
```



```
ggplot(melted.categorial.control, aes(x=value, y=MD_EARN_WNE_P10, fill=value)) +
  geom_boxplot() +
  geom_jitter(color="black", size=0.4, alpha=0.9) +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("Public vs. Private") +
  xlab("CONTROL") +
  scale_x_discrete(labels = c("Public","Private","Non-profit private"))
```



```
ggplot(melted.categorial.locale, aes(x=value, y=MD_EARN_WNE_P10, fill=value)) +
  geom_boxplot() +
  geom_jitter(color="black", size=0.4, alpha=0.9) +
  theme(
    legend.position="none",
    plot.title = element_text(size=11)
  ) +
  ggtitle("Location of school") +
  xlab("LOCALE") +
  labs(caption = "City (11-13), Suburb (21-23), Town (31-33), Rural (41-43)")
```


Location of school

City (11−13), Suburb (21−23), Town (31−33), Rural (41−43)

Phase 3 - Feature selection with stepwise regression

In the third phase, the remaining variables were used with stepwise regression, to test which subset of features perform the best, whether the received coefficients are reasonable, and if there are signs of overfitting.

The Stepwise regression suggested using all of the variables, except CITY and DOCTORAL. The stepwise regression can be seen in appendix 1. Because stepwise regression suggested leaving out DOCTORAL, for the sake of consistency we decided to also leave out MASTER as it was a the middle level in our institutional classifications and wouldn't have made much sense on its own.

The final features are listed in the table below:

	Name	*Data type*	*Description*
1	*SAT_ALL*	*float*	*Midpoint of SAT scores at the institution (critical reading , math)*
2	*MD_FAMINC*	*float*	*Median family income*
3	*AGE_ENTRY*	*float*	*Average age of entry*
4	*COSTT4_A*	*float*	*Average cost of attendance (academic year institutions)*
5	*POVERTY_RATE*	*float*	*Poverty rate*
6	*PRIVATE*	*binary*	*Carnegie Classification − −basic*

And the dependent variable:

Name	*Data type*	*Description*
MD_EARN_WNE_P10	*float*	*Median earnings of students 10 years after entry*

Generating binary features from the categorical features

```
data.joined.model <- data.joined.dropna %>%
  mutate(URBAN = case_when(LOCALE %in% c(seq(11,13), seq(21,23)) ~ 1,
                           TRUE ~ 0),
         PRIVATE = case_when(CONTROL %in% c(2,3) ~ 1,
                             TRUE ~ 0),
         DOCTORAL = case_when(CCBASIC %in% seq(15,17) ~ 1,
                              TRUE ~ 0),
         MASTER = case_when(CCBASIC %in% seq(18,20) ~ 1,
                            TRUE ~ 0)
  )


numerical.vars.model <- c("MD_EARN_WNE_P10", "SAT_ALL", "MD_FAMINC", "AGE_ENTRY",
                          "COSTT4_A", "POVERTY_RATE")
categorical.vars.model <- c("URBAN", "PRIVATE", "DOCTORAL", "MASTER")

# data with REGION identifier for STAN
data.joined.stan <- data.joined.model %>%
  select(REGION, numerical.vars.model, categorical.vars.model)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(numerical.vars.model)‘ instead of ‘numerical.vars.model‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(categorical.vars.model)‘ instead of ‘categorical.vars.model‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

```
data.joined.stan.test <- data.joined.stan %>%
  sample_n(30)

data.joined.stan <- data.joined.stan %>%
  setdiff(data.joined.stan.test)

# data for linear regression model in R
data.joined.model <- data.joined.stan %>%
    select(-REGION)
```

3. Description of the models

MATEMATIIKKA TARKISTETTAVA

Description of the separate model

In the separate model, posteriors for the parameters are constructed. In the context of the project, the separate model considers all regions independent from each other, meaning that each region have individual parameters (mu, sigma).

Mathematical description:

$y_{ij}|\mu_j, \sigma_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$

where $\mu_j = \alpha_j + \beta_{\mathbf{j}}\mathbf{X}$

The parameters of the parameter vector are given in the section 4 with their priors.

Description of the pooled model

As in the separate model, the pooled model constructs posteriors for the parameters. However, the regions are considered as one entity meaning that all regions share the same distribution and parameter values.

Mathematical description:

$y_i|\mu, \sigma \sim \mathcal{N}(\mu, \sigma^2)$

where $\mu = \alpha + \beta\mathbf{X}$

The parameters of the parameter vector are given in the section 4 with their priors.

Description of the hierarchical model

Contrary to the other two models, in the hierarchical model posteriors are constructed for the prior parameters. With the hierarchical model, the regions are considered as individual but similar. In the context of the project, the regions share same sigma and the parameters forming mu are similarly distributed (sharing the hyperparameters).

Mathematical description:

$y_{ij}|\mu_j \sim \mathcal{N}(\mu_j, \sigma^2)$

where $\mu_j|\mu_P, \sigma_P^2 \sim \mathcal{N}(\mu_P, \sigma_P^2)$

The hyperparameters and parameters are given in the section 4 with their priors.

Description of the linear model

4. Informative or weakly informative priors, and justification of their choices.

We use weakly informative priors, as we do not posses enough information about the dependency between the independent variables and the dependent variable. When selecting the priors, proper distribution type was considered for each variable. The prior standard deviations were selected based on the magnitude of absolute values of the variables and what kind of impact they could have for income, with loose enough estimates to avoid limiting the values too much. Alternative approach could have been to standardize all variables, which would have reduced the need to think about magnitudes of absolute values.

SAT_ALL ~ Normal(43, 500)

Justification: We agreed that it was somewhat reasonable to expect scores in the SAT exam to be associated with higher income. Due to our approach of using weakly informative priors, we set a high standard deviation, but set a positive mean due to expected positive association.

The mean is calculated with the following formula: Median individual income in the United States / Average SAT Score (math and writing).

The median individual income in the US was approximately 31 000 in 2020 (Data Commons, 2020). The average SAT score in the US in 2020 were 523 in Math and 582 in Evidence-Based Reading and Writing (Number2, 2020).

mu = 31 000 / (523 + 528/2) = 43.2

MD_FAMINIC ~ Normal(0, 100)

Justification: Weakly informative prior is again selected as we don't posses enough information on the dependency. The absolute values are in general high, (for example compared to AGE_OF_ENTRY) and thus the standard deviation is set lower for this variable. However, there could be cases where MD_FAMINIC is low, due to for example unemployment, so the standard deviation is still set considerably high.

Ehdotus

This is a variable we will try to transform to logarithmic scale as impact of say, 10 000$ more income can be far greater for a low-income family than for a high-income family.

AGE_ENTRY ~ Normal(0, 2500)

Justification: We don't have a strong expectation of direction or magnitude of effect on income. As the age of entry could be somewhere in the range of 15 - 50 without considering outliers, a few dozen years difference could have dramatic changes in income either way, the standard deviation is set high.

COSTT4_A ~ Normal(0, 500)

Justification: Weakly informative prior is selected as we don't posses enough information on the dependency. The average cost per academic year is likely to take lower values than median family income, but higher values than average age of entry and thus the standard deviation is set between the standard deviations of those.

POVERTY_RATE ~ Normal(0, 2500)

Tää ei oikein täsmää. Jos tää sais ison painon pienillä arvoilla niin se tarkottais että kun liikutaan esim 2% köyhyydestä 3% köyhyyteen niin on tosi isot vaikutukset tuloihin. Todellisuudessa alueet on varmaan aika samanlaisia hyvinvoivia alueita. Nollaköyhyys on intercept-arvo josta sitten lähetään askeleittain kattomaan että miten tulot muuttuu kun alueen köyhyys lisääntyy. RATKAISUEHDOTUS: Lasketaan sd:tä. Ei tarvii olla yhtä dramaattinen kuin binäärisillä muutujilla.

Justification: The possible values are between 0 and 100. There could be situations were poverty rate in an area is really low, for instance 0.5%. For that reason, the standard deviation in the prior is set high to enable possibly high weight for a small value.

MASTER ~ Normal(0, 2500)

PRIVATE ~ Normal(0, 2500)

Justification: For MASTER and PRIVATE weakly informative prior, is selected as we don't posses enough information on the dependency with the dependent variable. Because the values are always either 0 and 1, the standard deviation is set high.

5. Stan, rstanarm or brms code.

Separate model

```
separate.model <- cmdstan_model(stan_file = "./Stan/separate.stan")

separate.model.data <- list(N = nrow(data.joined.stan),
                            K = length(unique(data.joined.stan$REGION)),
                            x = data.joined.stan$REGION,

                            SAT_ALL = data.joined.stan$SAT_ALL,
                            MD_FAMINIC = data.joined.stan$MD_FAMINC,
                            AGE_ENTRY = data.joined.stan$AGE_ENTRY,
                            COSTT4_A = data.joined.stan$COSTT4_A,
                            POVERTY_RATE = data.joined.stan$POVERTY_RATE,
                            PRIVATE = data.joined.stan$PRIVATE,
                            y = data.joined.stan$MD_EARN_WNE_P10,

                            pm_alpha = 0,
                            ps_alpha = 10000,
                            pm_SAT_ALL = 50,
                            ps_SAT_ALL = 500,
                            pm_MD_FAMINC = 0,
                            ps_MD_FAMINC = 100,
                            pm_AGE_ENTRY = 0,
                            ps_AGE_ENTRY = 2500,
                            pm_COSTT4_A = 0,
                            ps_COSTT4_A = 500,
                            pm_POVERTY_RATE = 0,
                            ps_POVERTY_RATE = 2500,
                            pm_PRIVATE = 0,
                            ps_PRIVATE = 2500,
                            pm_sigma = 10000,
                            ps_sigma = 10000

)


separate.fit <- separate.model$sample(data = separate.model.data, seed = 1234, refresh = 1e3)

separate.fit
```

Pooled model

```
pooled.model <- cmdstan_model(stan_file = "./Stan/pooled.stan")

pooled.model.data <- list(N = nrow(data.joined.stan),

                          SAT_ALL = data.joined.stan$SAT_ALL,
                          MD_FAMINIC = data.joined.stan$MD_FAMINC,
                          AGE_ENTRY = data.joined.stan$AGE_ENTRY,
                          COSTT4_A = data.joined.stan$COSTT4_A,
                          POVERTY_RATE = data.joined.stan$POVERTY_RATE,
                          PRIVATE = data.joined.stan$PRIVATE,
                          y = data.joined.stan$MD_EARN_WNE_P10,

                          pm_alpha = 0,
                          ps_alpha = 10000,
                          pm_SAT_ALL = 50,
                          ps_SAT_ALL = 500,
                          pm_MD_FAMINC = 0,
                          ps_MD_FAMINC = 100,
                          pm_AGE_ENTRY = 0,
                          ps_AGE_ENTRY = 2500,
                          pm_COSTT4_A = 0,
                          ps_COSTT4_A = 500,
                          pm_POVERTY_RATE = 0,
                          ps_POVERTY_RATE = 2500,
                          pm_PRIVATE = 0,
                          ps_PRIVATE = 2500,
                          pm_sigma = 10000,
                          ps_sigma = 10000

                          )


pooled.fit <- pooled.model$sample(data = pooled.model.data, seed = 1234, refresh = 1e3)

pooled.fit
```

Hierarchical model

```
hierarchical.model <- cmdstan_model(stan_file = "./Stan/hierarchical.stan")

hierarchical.model.data <- list(N = nrow(data.joined.stan),
                                K = length(unique(data.joined.stan$REGION)),
                                x = data.joined.stan$REGION,

                                SAT_ALL = data.joined.stan$SAT_ALL,
                                MD_FAMINIC = data.joined.stan$MD_FAMINC,
                                AGE_ENTRY = data.joined.stan$AGE_ENTRY,
                                COSTT4_A = data.joined.stan$COSTT4_A,
                                POVERTY_RATE = data.joined.stan$POVERTY_RATE,
                                PRIVATE = data.joined.stan$PRIVATE,
                                y = data.joined.stan$MD_EARN_WNE_P10,

                                pm_alpha = 0,
                                ps_alpha = 10000,
                                pm_s_alpha = 100,
                                ps_s_alpha = 100,
                                pm_SAT_ALL = 43,
                                ps_SAT_ALL = 500,
                                pm_s_SAT_ALL = 100,
                                ps_s_SAT_ALL = 100,

                                pm_MD_FAMINC = 0,
                                ps_MD_FAMINC = 100,
                                pm_s_MD_FAMINC = 10,
                                ps_s_MD_FAMINC = 100,

                                pm_AGE_ENTRY = 0,
                                ps_AGE_ENTRY = 2500,
                                pm_s_AGE_ENTRY = 500,
                                ps_s_AGE_ENTRY = 500,

                                pm_COSTT4_A = 0,
                                ps_COSTT4_A = 500,
                                pm_s_COSTT4_A = 50,
                                ps_s_COSTT4_A = 500,

                                pm_POVERTY_RATE = 0,
                                ps_POVERTY_RATE = 2500,
                                pm_s_POVERTY_RATE = 500,
                                ps_s_POVERTY_RATE = 500,

                                pm_PRIVATE = 0,
                                ps_PRIVATE = 2500,
                                pm_s_PRIVATE = 500,
                                ps_s_PRIVATE = 500,

                                pm_sigma = 10000,
                                ps_sigma = 10000,
                                pm_s_sigma = 500,
                                ps_s_sigma = 500)

hierarchical.fit <- hierarchical.model$sample(data = hierarchical.model.data,
                                              seed = 1234, refresh = 1e3)

hierarchical.fit
```

6. How to the Stan model was run, that is, what options were used.

This is also more clear as combination of textual explanation and the actual code line.

7. Convergence diagnostics (R^, ESS, divergences) and what was done if the convergence was not good with the first try.

Separate model Rhat

```r
rhat.df <- tibble()

params <- c("alpha[1]", "beta_SAT_ALL[1]", "beta_MD_FAMINIC[1]",
            "beta_AGE_ENTRY[1]", "beta_COSTT4_A[1]", "beta_POVERTY_RATE[1]",
            "beta_MASTER[1]", "beta_PRIVATE[1]")

for (param in params) {
  rhats <- extract_variable_matrix(separate.fit$draws(), variable = param) %>% apply(2, rhat)
  row <- tibble("Parameter" = param,
                "Chain 1" = rhats[1],
                "Chain 2" = rhats[2],
                "Chain 3" = rhats[3],
                "Chain 4" = rhats[4])
  rhat.df <- rbind(rhat.df, row)
}

rhat.df
```

Rhat diagnostics

We used the rhat() function for all models to get the rhat convergence diagnostic. The function compares the between chain and within chain estimates for model parameters and other univariate quantities of interest. If the between chain and within chain estimates agree, it is said that the chains have mixed well. This happens when R-hat is less than 1.05 or less than 1.01 depending on the standard. We decided not be strict with our multivariate model and chose the less strict threshold of 1.05.

Pooled model Rhat

```r
params <- pooled.model$variables()$parameters %>% names()
rhat.df <- tibble()
for (param in params) {
  rhats <- extract_variable_matrix(pooled.fit$draws(), variable = param) %>% apply(2, rhat)
  row <- tibble("Parameter" = param,
                "Chain 1" = rhats[1],
                "Chain 2" = rhats[2],
                "Chain 3" = rhats[3],
                "Chain 4" = rhats[4])
  rhat.df <- rbind(rhat.df, row)
}
rhat.df
```

Hierarchical model Rhat

```
rhat.df <- tibble()

params <- c("alpha[1]", "beta_SAT_ALL[1]", "beta_MD_FAMINIC[1]",
            "beta_AGE_ENTRY[1]", "beta_COSTT4_A[1]", "beta_POVERTY_RATE[1]",
            "beta_MASTER[1]", "beta_PRIVATE[1]")

for (param in params) {
  rhats <- extract_variable_matrix(hierarchical.fit$draws(), variable = param) %>% apply(2, rhat)
  row <- tibble("Parameter" = param,
                "Chain 1" = rhats[1],
                "Chain 2" = rhats[2],
                "Chain 3" = rhats[3],
                "Chain 4" = rhats[4])
  rhat.df <- rbind(rhat.df, row)
}

rhat.df
```

Effective sample size (ESS)

Effective sample size (ESS) evaluates the uncertainty in estimates that is caused by autocorrelation of the chains. The value of effective sample size represents the number of independent samples that poses the same predictive power as all the autocorrelated samples. In other words, if the number of effective sample size is high, then the number of independent samples is high.

As can be seen on the tables below, for the separate model the values for the first region are range from 54 to 96. In this region, there are the sample size N is xxx. This means that…

For the pooled model the ESS values for parameters range from 508 to 954. This means that most of the samples for all parameters are independent, although there is also autocorrelation in most cases. According to (Stan) if the effective sample size is larger than the number of samples, this can be due to antithetic Markov Chains which have negative autocorrelations on odd lags.

For the hierarchical model the ESS values for parameters are between 266 and 1080. This means that for some parameters there samples are more autocorrelated samples than independent ones. The analysis for the highest values is the same as for pooled.

Lähde: https://mc-stan.org/docs/2_19/reference-manual/effective-sample-size-section.html

Serparate Model Effective sample size (ESS)

```
params <- separate.model$variables()$parameters %>% names()
ess.df <- tibble()

params <- c("alpha[1]", "beta_SAT_ALL[1]", "beta_MD_FAMINIC[1]",
            "beta_AGE_ENTRY[1]", "beta_COSTT4_A[1]", "beta_POVERTY_RATE[1]",
            "beta_MASTER[1]", "beta_PRIVATE[1]")

for (param in params) {
  ess <- extract_variable_matrix(separate.fit$draws(), variable = param) %>% apply(2, ess_basic)
  row <- tibble("Parameter" = param,
                "ESS" = ess)
  ess.df <- rbind(ess.df, row)
}
ess.df <- ess.df  %>% group_by(Parameter) %>% summarise(ESS = sum(ESS)/n(),)

ess.df
```

Pooled Model Effective sample size (ESS)

```
#pooled.fit$cmdstan_summary()

params <- pooled.model$variables()$parameters %>% names()
ess.df <- tibble()

for (param in params) {
  ess <- extract_variable_matrix(pooled.fit$draws(), variable = param) %>% apply(2, ess_basic)
  row <- tibble("Parameter" = param,
                "ESS" = ess)
  ess.df <- rbind(ess.df, row)
}
ess.df <- ess.df  %>% group_by(Parameter) %>% summarise(ESS = sum(ESS)/n(),)

ess.df
```

Hierarchical Model Effective sample size (ESS)

```
params <- hierarchical.model$variables()$parameters %>% names()
ess.df <- tibble()

params <- c("alpha[1]", "beta_SAT_ALL[1]", "beta_MD_FAMINIC[1]",
            "beta_AGE_ENTRY[1]", "beta_COSTT4_A[1]", "beta_POVERTY_RATE[1]",
            "beta_MASTER[1]", "beta_PRIVATE[1]")

for (param in params) {
  ess <- extract_variable_matrix(hierarchical.fit$draws(), variable = param) %>% apply(2, ess_basic)
  row <- tibble("Parameter" = param,
                "ESS" = ess)
  ess.df <- rbind(ess.df, row)
}
ess.df <- ess.df  %>% group_by(Parameter) %>% summarise(ESS = sum(ESS)/n(),)

ess.df
```

HMC specific convergence diagnostics for all models

Based on the HMC specific convergence diagnostics, the separate model reached the initial maximum treedepth of 10 in 100% of the transitions. The diagnostics state that this leads to premature termination of trajectories and slow exploration. The proposed action of increasing the limit was tested, but it resulted to too long running time for the already time consuming fitting of the separate model. The rest of the diagnostics consisting of divergences, E-BMFI, effective sample size and split R-hat satisfactory.

For the pooled model, treedepth, divergences, E-BFMI, effective sample size and split R-hat were satifactory.

For the hierarchical model, onyl 4 out of 4000 hit the maximum treedepth and 23 out of 4000 transitions did not converge. Due to relatively low numbers (0.1% and 0.57%) no actions were taken for the hierarchical model. The rest of the diagnostics were satisfactory.

```r
separate.fit$cmdstan_diagnose()

pooled.fit$cmdstan_diagnose()

hierarchical.fit$cmdstan_diagnose()
```

8. Posterior predictive checks and model comparison

Posterior predictive checking

When doing posterior predictive checking we look for systematic discrepancies between the real observations and the data we get from simulating replicated data under the fitted model (Gelman and Hill, 2006). Posterior predictive checking is a form of internal validation that is a helpful phase of model building and checking in assessing whether our fitted model makes sense.

```r
separate.extract <- separate.fit$draws(variable = "y_rep")

y <- data.joined.stan$MD_EARN_WNE_P10

y_rep <- matrix(data = separate.extract[,1,], nrow = 1000)

separate.ppctitle <- ggtitle("Separate model",
                             "Comparing densities of y and y_rep")

ppc_dens_overlay(y, y_rep) + separate.ppctitle

# Pooled

pooled.extract <- pooled.fit$draws(variable = "y_rep")

y_rep <- matrix(data = pooled.extract[,1,], nrow = 1000)

pooled.ppctitle <- ggtitle("Pooled model",
                             "Comparing densities of y and y_rep")

ppc_dens_overlay(y, y_rep) + pooled.ppctitle

# Hierarchical

hierarchical.extract <- hierarchical.fit$draws(variable = "y_rep")

y_rep <- matrix(data = hierarchical.extract[,1,], nrow = 1000)

hierarchical.ppctitle <- ggtitle("Hierarchical model",
                             "Comparing densities of y and y_rep")

ppc_dens_overlay(y, y_rep) + hierarchical.ppctitle
```

LOO

The ELPD is the theoretical expected log pointwise predictive density for a new dataset (Eq 1 in VGG2017), which can be estimated, e.g., using cross-validation. elpd_loo is the Bayesian LOO estimate of the expected log pointwise predictive density (Eq 4 in VGG2017) and is a sum of N individual pointwise log predictive densities. Probability densities can be smaller or larger than 1, and thus log predictive densities can be negative or positive. For simplicity the ELPD acronym is used also for expected log pointwise predictive probabilities for discrete models. Probabilities are always equal or less than 1, and thus log predictive probabilities are 0 or negative.

LOO (Model comparison)

Yläpuolella kopsattu, kirjoita uusiks. Lähde alla

Lähde: https://mc-stan.org/loo/reference/loo-glossary.html

With $elpd_{loo-cv}$ we should select the model with the highest value. From the tables below we can see that in this case the suggested model would be separate model, the second best hierarchical model and the third best pooled model.

```
separate.loo <- separate.fit$loo()

pooled.loo <- pooled.fit$loo()

hierarchical.loo <- hierarchical.fit$loo()

loo_compare(separate.loo, pooled.loo, hierarchical.loo)

separate.loo

pooled.loo

hierarchical.loo
```

```
loo_compare(separate.loo, pooled.loo, hierarchical.loo)
```

K hat

```
pareto_k_table(separate.fit$loo())

pareto_k_table(pooled.fit$loo())

pareto_k_table(hierarchical.fit$loo())
```

```
plot(separate.fit$loo(), main = "Separate model PSIS diagnostics")

plot(pooled.fit$loo(), main = "Pooled model PSIS diagnostics")

plot(hierarchical.fit$loo(), main = "Hierarchical model PSIS diagnostics")
```

Based on $\hat{K}$ the predictions are … too optimistic/reliable? - Why pooled model looks like that - overfitting or something?

9. Optional/Bonus: Predictive performance assessment if applicable (e.g. classification accuracy) and evaluation of practical usefulness of the accuracy. This should be reported for all models as well.

Root Mean Squared Error (RMSE)

Pooled model

```
pooled.fit.coeff.df <- pooled.fit$summary() %>% slice(2:8)

alpha.hat <- pooled.fit.coeff.df %>% slice(1) %>% select(median)
beta.hat <- pooled.fit.coeff.df %>% slice(2:8) %>% select(median) %>% as.matrix()

test.X <- data.joined.stan.test %>%
  select(SAT_ALL, MD_FAMINC, AGE_ENTRY, COSTT4_A, POVERTY_RATE, PRIVATE) %>%
  as.matrix()

N <- nrow(data.joined.stan.test)
y.hat <- numeric(N)
y <- data.joined.stan.test$MD_EARN_WNE_P10
se <- numeric(N)
for (i in 1:N) {

  y.hat[i] <- alpha.hat+beta.hat%*%test.X[i,]
  se[i] <- (y.hat[[i]]-y[i])^2

}

y.hat <- y.hat %>%
  as.matrix()
```

```
se %>% plot(type="b", pch=20)
```

```
rmse <- sqrt(mean(se))
rmse
```

10. Sensitivity analysis with respect to prior choices (i.e. checking whether the result changes a lot if prior is changed). This should be reported for all models.

To test whether our results are sensitive to prior choices, and to see how changing the priors affects our results, we ran all the models with both wider and narrower priors. The wide priors were obtained by multiplying each standard deviation by three, and the narrow priors were obtained by dividing the original priors by two.

11. Discussion of issues and potential improvements.

During the project, the large amount of variables also posed challenges, as it was arguably rather slow and burdensome to find the most relevant variables to use in our analysis. It was somewhat surprising how fast the observation count started to shrink in data cleaning process, so in hindsight more attention could have been paid to cleanliness, as this dataset had for example a lot of missing values.

It was a shame we couldn't use a larger testing dataset because all data used for testing would be away from training the model. We tried to reduce the need to have a very long dataset by working very hard on removing independent variables. That work also ended up expanding our observation count from roughly 200 to over 700 observations, as we found variables with less null values and removed variables with too much missing data. For the feature selection, with more time we could have experimented with alternative methods, like the LASSO or Ridge regression.

As we were building a predictive model, we had to take certain ethical questions into account, especially to avoid building a discriminatory model. For example, if our model was used to by an employer to assess how well alumni from a certain school would perform in their career, it would be discriminatory to have a model that predicts lower performance for a university with say, high female or black student population as the real reason for differences in our data might be something else entirely than skin color or gender. In the end, this wasn't a big issue as for example the correlations between both gender vs income and share of white population vs income were very low and they would have been dropped from our model no matter what.

12. Conclusion what was learned from the data analysis.

13. Self-reflection of what the group learned while making the project.

In hindsight taking on a task of creating three multivariate models was very ambitious, because multivariable models had not been covered very much during the course. That ambition paid off, however, and we think we have now a much stronger grasp of how the different models - pooled, separate and hierarchical - work and also know how to build multivariate bayesian linear models in general. We also learned a lot about feature selection, because the 3000 variables in the dataset forced us to create sensible procedures for selection. The combination of using logic and feature engineering tools like stepwise regression is a very powerful skill we developed while making this project.

I think this project made us think really hard about what do we want to show the reader to make our work as understandable as possible and we learned to visualize data and results in interesting ways and apply the visualization techniques covered on the course. Making the visualizations also aided our own thinking and certain visualizations gave valuable insight on our results and their quality.

Working with Stan is something none of us had done before this course. Stan is clearly a very powerful tool for statistical and data science use and we feel this project really ingrained basic Stan workflows and made working with Stan feel more of an efficient routine than an obstacle.

14. References

Card, D. (1999). THE CAUSAL EFFECT OF EDUCATION ON EARNINGS. Wolla, S. A., & Sullivan, J. (2017). Education, Income, and Wealth. https://fred.stlouisfed.org/graph/?g=7yKu.

Data Commons. (2020). Gross domestic product per capita in United States of America [Graph]. Retrieved from https://datacommons.org/place/country/USA?utm_medium=explore&mprop=income&popt=Person&cpv=age%2CYears15Onwards&hl=en

Gelman, Andrew, and Jennifer Hill. Data analysis using regression and multilevel/hierarchical models. Cambridge university press, 2006.

Number2. (2020). Average SAT Score [Blog Post]. Retrieved from: https://www.number2.com/average-sat-score/

15. Appendices

Appendix 1 - Stepwise regression

```r
# PRELIMINARY ANALYSIS ----
# preliminary model with all numerical vars (not yet categorical)
# MODELING ----

data.joined.model <- data.joined.dropna %>%
  mutate(URBAN = case_when(LOCALE %in% c(seq(11,13), seq(21,23)) ~ 1,
                           TRUE ~ 0),
         PRIVATE = case_when(CONTROL %in% c(2,3) ~ 1,
                             TRUE ~ 0),
         DOCTORAL = case_when(CCBASIC %in% seq(15,17) ~ 1,
                              TRUE ~ 0),
         MASTER = case_when(CCBASIC %in% seq(18,20) ~ 1,
                            TRUE ~ 0)
  )


numerical.vars.model <- c("MD_EARN_WNE_P10", "SAT_ALL", "MD_FAMINC", "AGE_ENTRY",
                          "COSTT4_A", "POVERTY_RATE")
categorical.vars.model <- c("URBAN", "PRIVATE", "DOCTORAL", "MASTER")

# data with REGION identifier for STAN
data.joined.stan <- data.joined.model %>%
  select(REGION, numerical.vars.model, categorical.vars.model)

# data for linear regression model in R
data.joined.model <- data.joined.stan %>%
    select(-REGION)
```

```r
# baseline model
model <- lm(MD_EARN_WNE_P10 ~ ., data = data.joined.model)
summary(model)
```

```
##
## Call:
## lm(formula = MD_EARN_WNE_P10 ~ ., data = data.joined.model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21261.2  -4348.1   -879.1   3665.5  31049.9
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.284e+04  4.825e+03  -2.661 0.007946 **
## SAT_ALL       9.444e+01  5.901e+00  16.005  < 2e-16 ***
## MD_FAMINC     4.303e-02  2.019e-02   2.131 0.033362 *
## AGE_ENTRY    -1.192e+01  1.409e+02  -0.085 0.932603
## COSTT4_A      3.823e-01  3.409e-02  11.212  < 2e-16 ***
## POVERTY_RATE -2.011e+02  7.838e+01  -2.566 0.010469 *
## URBAN         2.172e+03  5.666e+02   3.834 0.000136 ***
## PRIVATE      -8.111e+03  9.801e+02  -8.276 5.44e-16 ***
## DOCTORAL      4.038e+02  8.149e+02   0.495 0.620389
## MASTER        1.058e+03  6.924e+02   1.529 0.126790
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 6616 on 786 degrees of freedom
## Multiple R-squared:  0.6777, Adjusted R-squared:  0.674
## F-statistic: 183.6 on 9 and 786 DF,  p-value: < 2.2e-16
```

```r
# step wise regression implied "best" model in terms of AIC
step(model, direction = "backward")
```

```
## Start:  AIC=14015.21
## MD_EARN_WNE_P10 ~ SAT_ALL + MD_FAMINC + AGE_ENTRY + COSTT4_A +
##     POVERTY_RATE + URBAN + PRIVATE + DOCTORAL + MASTER
##
##                 Df  Sum of Sq        RSS   AIC
## - AGE_ENTRY      1 3.1328e+05 3.4407e+10 14013
## - DOCTORAL       1 1.0747e+07 3.4418e+10 14014
## <none>                        3.4407e+10 14015
## - MASTER         1 1.0227e+08 3.4509e+10 14016
## - MD_FAMINC      1 1.9887e+08 3.4606e+10 14018
## - POVERTY_RATE   1 2.8825e+08 3.4695e+10 14020
## - URBAN          1 6.4344e+08 3.5050e+10 14028
## - PRIVATE        1 2.9983e+09 3.7405e+10 14080
## - COSTT4_A       1 5.5026e+09 3.9910e+10 14131
## - SAT_ALL        1 1.1213e+10 4.5620e+10 14238
##
## Step:  AIC=14013.22
## MD_EARN_WNE_P10 ~ SAT_ALL + MD_FAMINC + COSTT4_A + POVERTY_RATE +
##     URBAN + PRIVATE + DOCTORAL + MASTER
```

```
##
##                 Df  Sum of Sq        RSS   AIC
## - DOCTORAL      1 1.0445e+07 3.4418e+10 14012
## <none>                       3.4407e+10 14013
## - MASTER        1 1.0407e+08 3.4511e+10 14014
## - MD_FAMINC     1 2.7912e+08 3.4686e+10 14018
## - POVERTY_RATE  1 2.9723e+08 3.4704e+10 14018
## - URBAN         1 6.4550e+08 3.5053e+10 14026
## - PRIVATE       1 3.3040e+09 3.7711e+10 14084
## - COSTT4_A      1 5.6436e+09 4.0051e+10 14132
## - SAT_ALL       1 1.1213e+10 4.5620e+10 14236
##
## Step:  AIC=14011.46
## MD_EARN_WNE_P10 ~ SAT_ALL + MD_FAMINC + COSTT4_A + POVERTY_RATE +
##     URBAN + PRIVATE + MASTER
##
##                 Df  Sum of Sq        RSS   AIC
## <none>                       3.4418e+10 14012
## - MASTER        1 1.1892e+08 3.4537e+10 14012
## - MD_FAMINC     1 2.7162e+08 3.4689e+10 14016
## - POVERTY_RATE  1 3.0210e+08 3.4720e+10 14016
## - URBAN         1 7.2639e+08 3.5144e+10 14026
## - PRIVATE       1 3.6055e+09 3.8023e+10 14089
## - COSTT4_A      1 5.7805e+09 4.0198e+10 14133
## - SAT_ALL       1 1.2613e+10 4.7030e+10 14258
```

```
##
## Call:
## lm(formula = MD_EARN_WNE_P10 ~ SAT_ALL + MD_FAMINC + COSTT4_A +
##     POVERTY_RATE + URBAN + PRIVATE + MASTER, data = data.joined.model)
##
## Coefficients:
##  (Intercept)       SAT_ALL     MD_FAMINC      COSTT4_A  POVERTY_RATE
##   -1.341e+04     9.533e+01     4.313e-02     3.846e-01    -2.012e+02
##        URBAN       PRIVATE        MASTER
##    2.240e+03    -8.246e+03     8.218e+02
```

```
stepwise.model <- lm(formula = MD_EARN_WNE_P10 ~ SAT_ALL + MD_FAMINC + COSTT4_A +
                       POVERTY_RATE + URBAN + PRIVATE + MASTER, data = data.joined.model)
summary(stepwise.model)
```

```
##
## Call:
## lm(formula = MD_EARN_WNE_P10 ~ SAT_ALL + MD_FAMINC + COSTT4_A +
##     POVERTY_RATE + URBAN + PRIVATE + MASTER, data = data.joined.model)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -21303.3  -4335.7   -872.9   3705.7  30767.8
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.341e+04  2.960e+03  -4.531 6.78e-06 ***
## SAT_ALL       9.533e+01  5.610e+00  16.993  < 2e-16 ***
## MD_FAMINC     4.313e-02  1.729e-02   2.494  0.01284 *
## COSTT4_A      3.846e-01  3.343e-02  11.504  < 2e-16 ***
## POVERTY_RATE -2.012e+02  7.652e+01  -2.630  0.00871 **
## URBAN         2.240e+03  5.492e+02   4.078 5.00e-05 ***
## PRIVATE      -8.246e+03  9.076e+02  -9.086  < 2e-16 ***
## MASTER        8.218e+02  4.980e+02   1.650  0.09934 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6609 on 788 degrees of freedom
## Multiple R-squared:  0.6776, Adjusted R-squared:  0.6747
## F-statistic: 236.6 on 7 and 788 DF,  p-value: < 2.2e-16
```