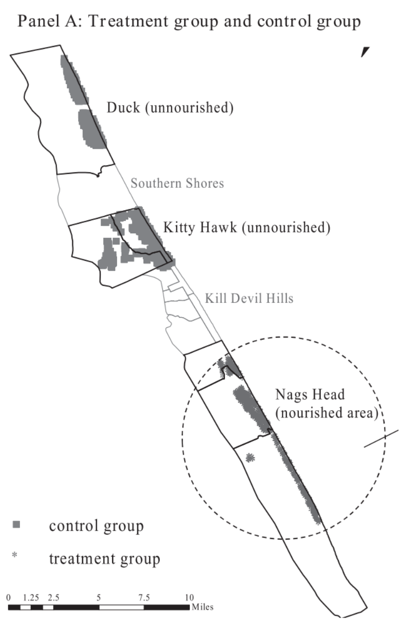
Replication Project

Blain Morin

10/30/2020

Background and Overview

This project replicates the difference-in-differences model estimated in Qiu and Gopalakrishnan's 2018 paper, "Shoreline defense against climate change and capitalized impact of beach nourishment" using a Bayesian framework (link to paper). The paper evaluates the impact of beach nourishment (which is "the process of replacing an eroding section of a beach with sand dredged from inlets or offshore sand reserves") on coastal housing prices. The analysis focuses on islands in the Outer Banks of North Carolina. Here is the map of the area analyzed for reference:



In the analysis, the authors compare the trend in sale prices of homes in Nags Head (the town that received beach nourishment) to the trend of sale prices in Duck and Kitty Hawk (which are towns that didn't receive beach nourishment until a few years after Nags Head). Here, receiving the beach nourishment is considered the treatment. The authors' diff-in-diff model takes the form:

$$ln(P_{ijt}) = \alpha_0 + \alpha_1 X_i + \beta_1 Nourish + \beta_2 PostNourish + \beta_3 Nourish * PostNourish + \eta_j + \zeta_t + \epsilon_{ijt}$$

where P_{ijt} is the price of home i in location j sold in year t. X_i is a vector of control variables for housing characteristics. $Nourish$ is an indicator variable for being in the treatment group. $PostNourish$ is an indicator that an observation is post treatment. The main coefficient of interest is β_3, which is the average treatment effect of beach nourishment. The authors also include census block group and year fixed effects (η_j and ζ_t respectively).

OLS Estimation Results

In the published paper, the authors use year fixed effects. Including the year fixed effect causes collinearity problems when I try to replicate their analysis (in other words the year fixed effects are collinear with the $PostNorish$ covariate, which is an indicator variable that = 1 when the year is greater than 2010). I also tried to filter the data based on the authors description, but am not able to exactly match their sample size. OLS regression results are presented in Table 1.

Table 1: OLS Regression Results

	Dependent variable:
	ln(Sale Price)
# of Bathrooms	0.054^{***} (0.009)
Living Area (100 Sqft)	0.040^{***} (0.003)
Living Area Squared	-0.0004^{***} (0.0001)
Age of Property (Years)	-0.005^{***} (0.0004)
# of Stories	0.091^{***} (0.013)
Distance to Shoreline (10m)	-0.002^{***} (0.0002)
Ocean Front = Yes	0.330^{***} (0.025)
Nourish	-0.135 (0.134)
PostNourish	-0.049^{***} (0.013)
Nourish*PostNourish	-0.004 (0.022)
Constant	12.074^{***} (0.057)
Location Fixed Effects?	Yes
Year Fixed Effects?	No
Observations	2,095
R^2	0.747
Adjusted R^2	0.744
Residual Std. Error	0.230 (df = 2074)
F Statistic	305.602^{***} (df = 20; 2074)
Note:	*p<0.1; **p<0.05; ***p<0.01

Bayesian Estimation Results

In this section, I replicate the analysis using rstan. Here is the specification in the .stan file:

```
data {
  int<lower=0> N;
  int<lower=1> K;
  matrix[N,K] x;
  vector[N] y;
}

parameters {
  vector[K] beta;
  real<lower=0> sigma;
}

model {
  y ~ normal(x * beta, sigma);
}
```

This model was run with 4 chains, each with 4000 iterations. A flat prior was used because of lack of prior information. The x matrix contains all of the housing covariates as well as location fixed effects. Table 2 presents the summary results for the non-fixed effect variables.

Table 2: Bayesian Estimation Results

term	mean	sd	n_eff	Rhat
Constant	12.0744	0.0568	2429.586	1.0004
# of Bathrooms	0.0537	0.0089	7334.114	1.0003
Living Area (100 Sqft)	0.0396	0.0031	6928.108	0.9999
Living Area Squared	-0.0004	0.0001	6438.678	1.0000
Age of Property (Years)	-0.0046	0.0004	7807.745	1.0000
# of Stories	0.0908	0.0126	6784.899	0.9997
Distance to Shoreline (10m)	-0.0019	0.0002	9085.058	1.0000
Ocean Front = Yes	0.3300	0.0255	6598.649	0.9998
Nourish	-0.1348	0.1353	2255.250	1.0015
PostNourish	-0.0494	0.0134	6139.730	1.0003
Nourish*PostNourish	-0.0042	0.0223	5977.502	1.0004
Sigma	0.2298	0.0035	6754.194	1.0005
lp__	2033.2651	3.3415	3119.229	1.0005

Looking at the summary results in Table 2, the effective sample and chain mixing (seen in the n_eff and Rhat columns, respectively) are sufficient for all variables. The estimated coefficients are also similar to the OLS specification.

Bayesian Diagnostics

Figure 1: MCMC Tracplots

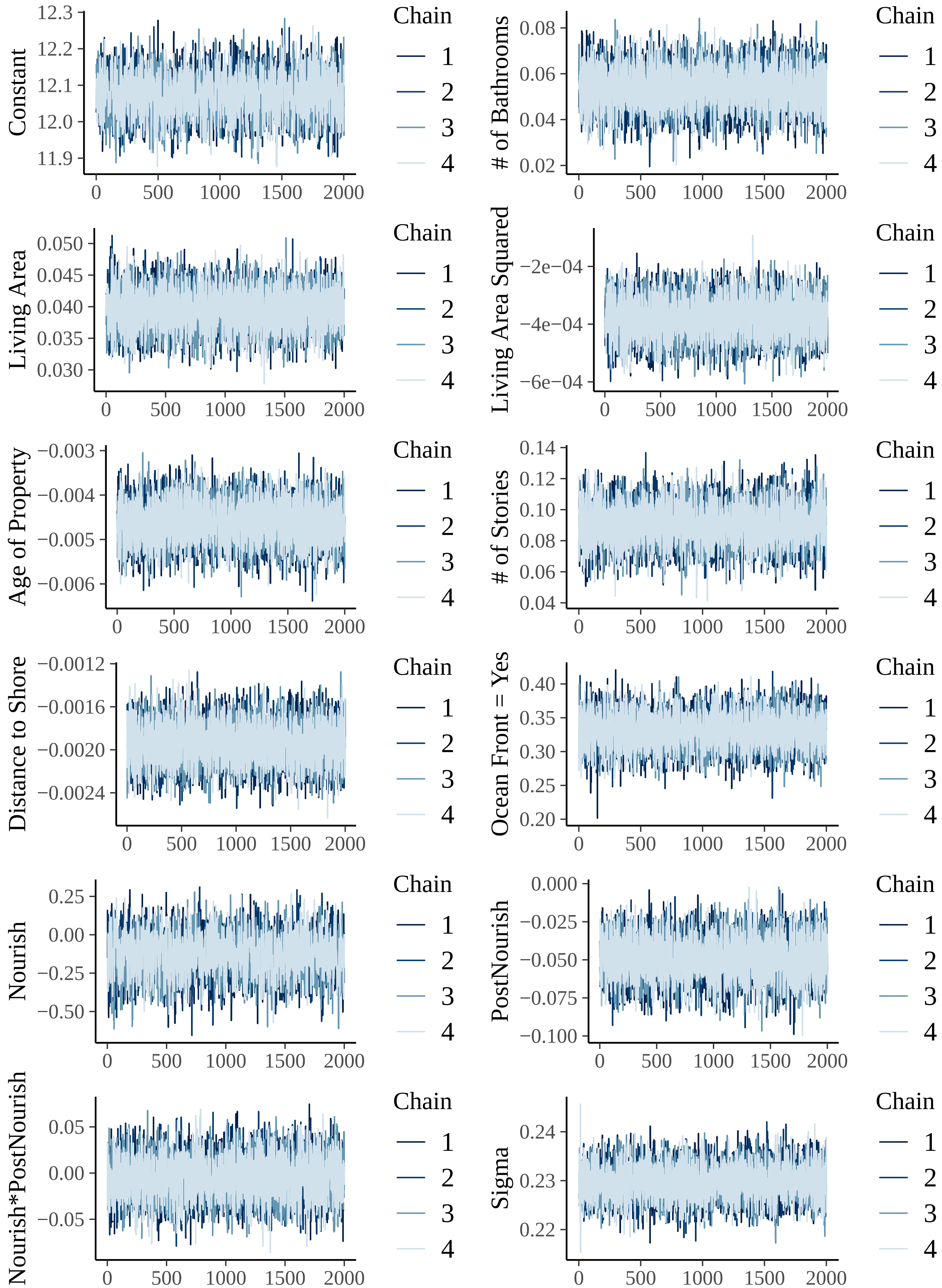
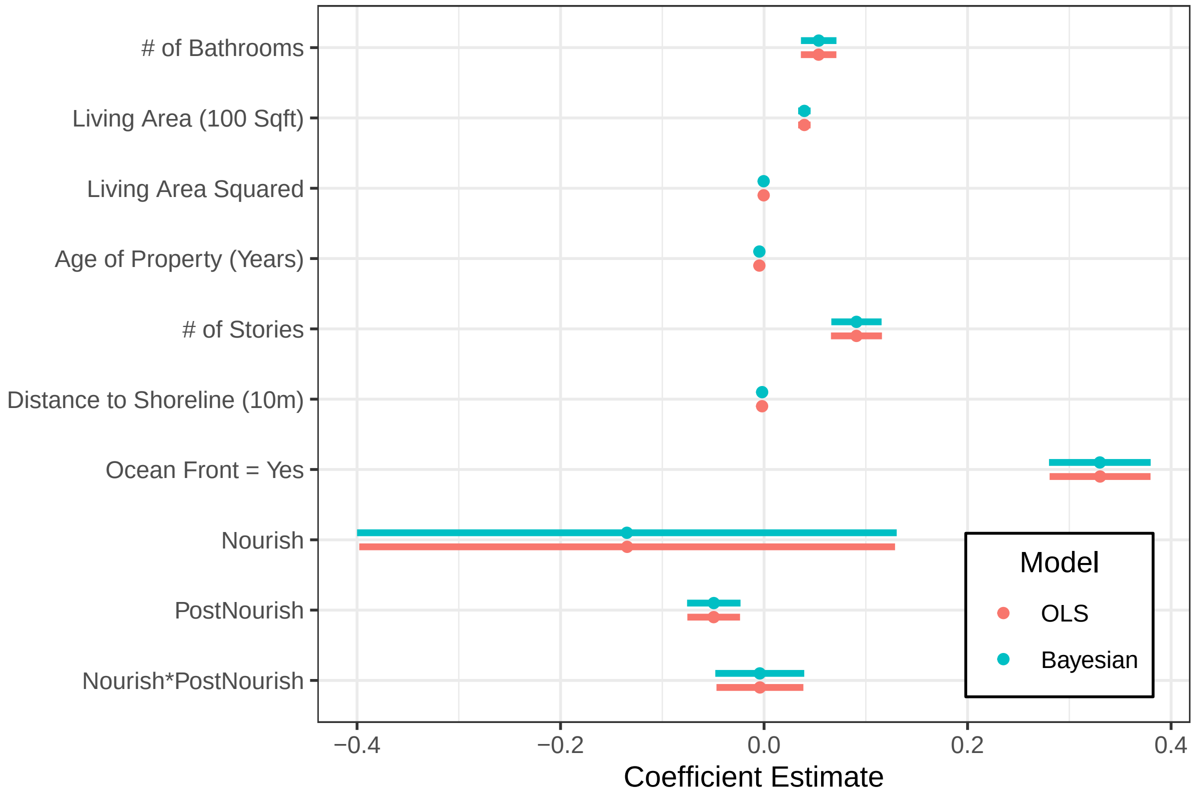


Figure 1 plots the MCMC chains for the variables of interest. As expected from the Rhat values in Table 1, the chains are mixing well and are bouncing around a narrow area.

Estimate Comparison

Figure 2: OLS versus Bayes Estimates



Overall, the model form in Qiu and Gopalakrishnan (2018) is relatively uncomplicated and starightforward. As seen in Figure 2, the confidence intervals and point estimates for each of the covariates are almost identical.

In Qiu and Gopalakrishnan's specification, they included year fixed effects. However, the post nourishment variable (which is an indicator for post treatment) is collinear with these year fixed effects. In Table 3, I show that the $PostNourish$ variable is perfectly predicted by these fixed effect variables. The model can run without producing warning using the lm function in R, if the dummy variables are not treated as factors. However, in the Bayesian framework, the MCMC chains do not mix and the number of effective samples drop dramatically. In sum, although the estimates are similar, on of the strengths of the Bayes approach is that it helps bring to light model misspecifocation.

Table 3: Collinearity Between Year Fixed Effects and Post-Nourishment

	Dependent variable:
	PostNourish1
sale_year2009	0.000***
	(0.000)
sale_year2010	0.000***
	(0.000)
sale_year2011	1.000***
	(0.000)
sale_year2012	1.000***
	(0.000)
sale_year2013	1.000***
	(0.000)
sale_year2014	1.000***
	(0.000)
Constant	−0.000***
	(0.000)
Observations	2,095
R^2	1.000
Adjusted R^2	1.000
Residual Std. Error	0.000 (df = 2088)
F Statistic	159,413,924,907,095,343,184,880,202,066.000*** (df = 6; 2088)
Note:	*p<0.1; **p<0.05; ***p<0.01

Appendix: Project Code

```r
knitr::opts_chunk$set(echo = FALSE)

### Load required packages

library(tidyverse)
library(rstan)
library(knitr)
library(dotwhisker)
library(broom)
library(bayesplot)
library(readxl)
library(gridExtra)
library(kableExtra)
library(stargazer)

### Read in data
```

```
df <- read_excel("Dare_geocoded.xls")


### Add map graphic
include_graphics("beach.PNG")

### Filter and clean the data according to authors' description
df1 = df %>%
  filter(sale_year >= 2008) %>%
  filter(sale_year <= 2014) %>%
  filter(dist_Shoreline_m < 2000) %>%
  mutate(Nourish = ifelse(city == "NAGS HEAD", 1, 0)) %>%
  mutate(PostNourish = ifelse(sale_year > 2010, 1, 0)) %>%
  mutate(age = sale_year - year_built) %>%
  mutate(sale_year = as.factor(sale_year)) %>%
  mutate(Nourish = as.factor(Nourish)) %>%
  mutate(PostNourish = as.factor(PostNourish))



### Specify the lm model
model1 = lm(log(price) ~ bathroom + sqft + I(sqft^2) + age + stry + I(dist_Shoreline_m / 10) + OceanFron


### Make a regression table
stargazer(model1, header = FALSE,
          title = "OLS Regression Results",
          single.row = TRUE,
          add.lines = list(c("Location Fixed Effects?", "Yes"), c("Year Fixed Effects?", "No")),
          omit = c("blkgrp"),
          dep.var.labels = "ln(Sale Price)",
          covariate.labels = c(
            "\\# of Bathrooms",
            "Living Area (100 Sqft)",
            "Living Area Squared",
            "Age of Property (Years)",
            "\\# of Stories",
            "Distance to Shoreline (10m)",
            "Ocean Front = Yes",
            "Nourish",
            "PostNourish",
            "Nourish*PostNourish"
          )
          )


data {
  int<lower=0> N;
  int<lower=1> K;
  matrix[N,K] x;
  vector[N] y;
}
```

```
parameters {
  vector[K] beta;
  real<lower=0> sigma;
}

model {
  y ~ normal(x * beta, sigma);
}
```

```
### Create the data list for stan
pricemodel = model.frame(log(price) ~ bathroom +
                           sqft + I(sqft^2) + age + stry +
                           I(dist_Shoreline_m / 10) + OceanFront + Nourish*PostNourish +
                           as.factor(blkgrp), data = df1)

price.data = list(y = model.response(pricemodel), x = model.matrix(pricemodel, pricemodel))
price.data$N = nrow(price.data$x)
price.data$K = ncol(price.data$x)


### Runs the final model specification
# options(mc.cores = parallel::detectCores())
# price.fit = stan("ols.stan", data = price.data, seed = 134, iter = 4000, chains = 4)
# save(price.fit, file = "finalmodel5")

load("finalmodel5")

price.fit.summary = round(summary(price.fit)$summary, 4)
price.fit.summary = as.data.frame(price.fit.summary)

term = c(colnames(price.data$x), "sigma", "lp__")

price.fit.summary = cbind(term, price.fit.summary)

df2 = price.fit.summary %>%
  select(term, mean, sd, n_eff, Rhat)

df2 = df2[c(1:10, 21:23),]

df3 = df2

df3$term = c(
            "Constant",
            "# of Bathrooms",
            "Living Area (100 Sqft)",
            "Living Area Squared",
            "Age of Property (Years)",
            "# of Stories",
            "Distance to Shoreline (10m)",
            "Ocean Front = Yes",
            "Nourish",
```

```
            "PostNourish",
            "Nourish*PostNourish",
            "Sigma",
            "lp_"
          )

### Makes a table of results

df3 %>%
  kable(booktabs = T, row.names = FALSE,
        linesep = "", caption = "Bayesian Estimation Results") %>%
  kable_styling(latex_options = "HOLD_position")


one.ten = c()

for (i in c(1:10, 21)) {

  one.ten[i] = paste("beta[", i, "]", sep = "")

}


ylabs = c(
            "Constant",
            "# of Bathrooms",
            "Living Area",
            "Living Area Squared",
            "Age of Property",
            "# of Stories",
            "Distance to Shore",
            "Ocean Front = Yes",
            "Nourish",
            "PostNourish",
            "Nourish*PostNourish",
            "Sigma",
            "lp_"
          )

a = mcmc_trace(price.fit, one.ten[1]) + ylab(ylabs[1])

b =mcmc_trace(price.fit, one.ten[2]) + ylab(ylabs[2])

c = mcmc_trace(price.fit, one.ten[3]) + ylab(ylabs[3])

d = mcmc_trace(price.fit, one.ten[4]) + ylab(ylabs[4])

e = mcmc_trace(price.fit, one.ten[5]) + ylab(ylabs[5])

f = mcmc_trace(price.fit, one.ten[6]) + ylab(ylabs[6])

g = mcmc_trace(price.fit, one.ten[7]) + ylab(ylabs[7])
```

```
h = mcmc_trace(price.fit, one.ten[8]) + ylab(ylabs[8])

i = mcmc_trace(price.fit, one.ten[9]) + ylab(ylabs[9])

j = mcmc_trace(price.fit, one.ten[10]) + ylab(ylabs[10])

k = mcmc_trace(price.fit, one.ten[21]) + ylab(ylabs[11])

l = mcmc_trace(price.fit, pars = "sigma") + ylab(ylabs[12])




grid.arrange(a, b, c, d, e, f, g, h, i, j, k, l, ncol = 2, top = "Figure 1: MCMC Tracplots")


m1_df = tidy(model1) %>%
  mutate(model = "OLS") %>%
  filter(!grepl("as.factor", term))

m1_df = m1_df[,-c(4,5)]

m1_df$term = c("Constant",
            "# of Bathrooms",
            "Living Area",
            "Living Area Squared",
            "Age of Property",
            "# of Stories",
            "Distance to Shore",
            "Ocean Front = Yes",
            "Nourish",
            "PostNourish",
            "Nourish*PostNourish"
          )

m1_df = m1_df %>%
  filter(term != "Constant")



m2_df = df3[-c(12, 13),]

rownames(m2_df) = NULL

m2_df = m2_df %>%
  rename(estimate = mean) %>%
  rename(std.error = sd) %>%
  mutate(model = "Bayesian") %>%
  select(-c(n_eff, Rhat)) %>%
  filter(term != "Constant")
```

```r
m1_df$term = m2_df$term

two_models = rbind(m1_df, m2_df)


dwplot(two_models, whisker_args = list(size = 1.2), dot_args = list(size = 1.5)) +
  theme_bw() +
  xlab("Coefficient Estimate") +
  ggtitle("Figure 2: OLS versus Bayes Estimates") +
  scale_color_discrete(name = "Model") +
  theme(legend.position = c(.85, .15),
        legend.background = element_rect(color = "black"),
        legend.title.align = .5)

pricemodel = model.frame(log(price) ~ bathroom +
                           sqft + I(sqft^2) + age + stry + I(dist_Shoreline_m / 10) +
                           OceanFront + Nourish*PostNourish + as.factor(blkgrp) + sale_year, data = df1)

price.data = list(y = model.response(pricemodel), x = model.matrix(pricemodel, pricemodel))


tester = as.data.frame(price.data$x)

tester = tester[, c(10, 21:26)]

tester2 = lm(PostNourish1 ~ ., data = tester)

stargazer(tester2, header = FALSE,
          title = "Collinearity Between Year Fixed Effects and Post-Nourishment")
```