output\_ryanstyle

# Import Packages + Read in Data

library(tidyverse)

## ── Attaching packages ─────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.3.0 ✓ purrr 0.3.3  
## ✓ tibble 3.0.0 ✓ dplyr 0.8.5  
## ✓ tidyr 1.0.0 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.4.0

## ── Conflicts ────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(magrittr)

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

## The following object is masked from 'package:tidyr':  
##   
## extract

library(here)

## here() starts at /Users/salmahuque/Desktop/BigThesis/ThesisR

library(data.table)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

## The following object is masked from 'package:purrr':  
##   
## transpose

library(MASS)

##   
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':  
##   
## select

library(knitr)  
library(stargazer)

##   
## Please cite as:

## Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

## R package version 5.2.2. https://CRAN.R-project.org/package=stargazer

library(kableExtra)

##   
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':  
##   
## group\_rows

library(lubridate)

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':  
##   
## hour, isoweek, mday, minute, month, quarter, second, wday, week,  
## yday, year

## The following object is masked from 'package:here':  
##   
## here

## The following object is masked from 'package:base':  
##   
## date

library(mapview)  
library(sf)

## Linking to GEOS 3.7.2, GDAL 2.4.2, PROJ 5.2.0

# read in csv  
thesis\_data <- read\_csv(here::here("DATA","thesis-data.csv"))

## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:  
## cols(  
## .default = col\_double(),  
## STATE\_ID = col\_character(),  
## RNO = col\_character(),  
## OWNER1 = col\_character(),  
## OWNER2 = col\_logical(),  
## OWNER3 = col\_logical(),  
## OWNERADDR = col\_character(),  
## OWNERCITY = col\_character(),  
## OWNERSTATE = col\_character(),  
## OWNERZIP = col\_character(),  
## SITEADDR = col\_character(),  
## SITECITY = col\_character(),  
## SITE\_STATE = col\_character(),  
## LEGAL\_DESC = col\_character(),  
## PROP\_CODE = col\_character(),  
## PRPCD\_DESC = col\_character(),  
## LANDUSE = col\_character(),  
## BEDROOMS = col\_logical(),  
## MKTVALYR3 = col\_character(),  
## SALEDATE = col\_character(),  
## ACC\_STATUS = col\_character()  
## # ... with 37 more columns  
## )

## See spec(...) for full column specifications.

## Warning: 19 parsing failures.  
## row col expected actual file  
## 8747 SHARED 1/0/T/F/TRUE/FALSE Y '/Users/salmahuque/Desktop/BigThesis/ThesisR/DATA/thesis-data.csv'  
## 20798 SHARED 1/0/T/F/TRUE/FALSE Y '/Users/salmahuque/Desktop/BigThesis/ThesisR/DATA/thesis-data.csv'  
## 20986 SHARED 1/0/T/F/TRUE/FALSE Y '/Users/salmahuque/Desktop/BigThesis/ThesisR/DATA/thesis-data.csv'  
## 21094 SHARED 1/0/T/F/TRUE/FALSE Y '/Users/salmahuque/Desktop/BigThesis/ThesisR/DATA/thesis-data.csv'  
## 21096 SHARED 1/0/T/F/TRUE/FALSE Y '/Users/salmahuque/Desktop/BigThesis/ThesisR/DATA/thesis-data.csv'  
## ..... ...... .................. ...... ..................................................................  
## See problems(...) for more details.

# Wrangling

# Split Data  
SFR <- thesis\_data %>%  
 filter(PRPCD\_DESC == "RESIDENTIAL IMPROVED", prop\_type == "Single-family")  
  
#-------------------------------------------------------------------------  
  
# Make a string of constraints and percent canopy names called "con\_names"  
clean <- SFR %>%  
 dplyr::select(contains("con"), contains("pct")) %>%  
 dplyr::select(-c("CONCRETE", "FIN SECOND", "PAVING/CONCRETE ONLY", "UNF SECOND", "conFldway", "pct\_conFldway"))   
clean\_names <- names(clean)  
con\_names <- paste(clean\_names, collapse = " + ")  
  
# switch the NAs in the constraints to 0s  
to0 <- function(x){ifelse(is.na(x), 0, x)}  
  
#-------------------------------------------------------------------------  
  
# Format SFR Dataframe:   
# Select variables, mutate filtering parameters. No actual filtering done at this point, all SFR vars kept.   
test1 <- SFR %>%  
 dplyr::select(-c(OWNER2, OWNER3, OWNERZIP,   
 MKTVALYR1, MKTVALYR2,  
 BLDGVAL1, BLDGVAL2,  
 LANDVAL1, LANDVAL2,   
 TOTALVAL1, TOTALVAL2,  
 MS\_GRADE, ES\_GRADE,  
 LEGAL\_DESC, TAXCODE,   
 PROP\_CODE, LANDUSE,   
 BEDROOMS, ACC\_STATUS,   
 NAME, COMMPLAN, SHARED,   
 COALIT, HORZ\_VERT, AUDIT\_NBRH,   
 MIDDLE\_SCH, Category, SOURCE,   
 FRONTAGE, COUNTY, YEARBUILT, bldgtype)) %>%  
 mutate(top\_1 = SALEPRICE > quantile(SALEPRICE, .99),  
 MKTVALYR3 = case\_when(MKTVALYR3 != 2018 ~ 2017,   
 TRUE ~ 2018),  
 price\_diff = SALEPRICE - LANDVAL3,   
 price\_ratio = SALEPRICE/LANDVAL3 \* 100,  
 vacant\_dummy = as.numeric(PRPCD\_DESC == "VACANT LAND"),  
 llc\_flag = grepl("LLC", OWNER1),  
 proud\_flag = grepl("PROUD", OWNER1),  
 trust\_flag = grepl("TRUST", OWNER1) & !grepl("FAMILY", OWNER1) & !grepl("LIVING", OWNER1))   
  
#-------------------------------------------------------------------------  
  
#garage sqft  
gar\_sqft\_sum <- test1 %>%  
 dplyr::select(matches("gar"), matches("car")) %>%  
 rowSums()  
  
#deck/patio/porch sqft  
deck\_sqft\_sum <- test1 %>%  
 dplyr::select(matches("deck"), matches("patio"), matches("porch")) %>%  
 rowSums()  
  
# attic sqft  
attic\_sqft\_sum <- test1 %>%  
 dplyr::select(matches("attic")) %>%  
 rowSums()  
  
# basement sqft  
bsmt\_sqft\_sum <- test1 %>%  
 dplyr::select(matches("bsmt")) %>%  
 dplyr::select(-c("BSMT PARKING","BSMT GAR")) %>%  
 rowSums()  
  
# creating zone change dummy  
zone\_test <- test1 %>%  
 dplyr::select(matches("zone"), -c(ZONE\_DESC\_aug2016, ZONE\_DESC\_aug2018, ZONE\_DESC\_feb2018, `Zone Description`, sale\_zone)) %>%  
 mutate\_all(str\_replace, pattern = "R", replacement = "") %>%  
 mutate\_all(funs(as.numeric)) %>%  
 rowSums()

## Warning: funs() is soft deprecated as of dplyr 0.8.0  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once per session.

## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion  
  
## Warning: NAs introduced by coercion

test1 %<>%  
 mutate(garage\_sqft = gar\_sqft\_sum,  
 garage\_dum = as.numeric(garage\_sqft > 0),  
 deck\_sqft = deck\_sqft\_sum,  
 deck\_dum = as.numeric(deck\_sqft > 0),  
 attic\_sqft = attic\_sqft\_sum,  
 attic\_dum = as.numeric(attic\_sqft > 0),  
 bsmt\_sqft = bsmt\_sqft\_sum,  
 bsmt\_dum = as.numeric(bsmt\_sqft > 0),  
 n\_fireplaces = replace\_na(n\_fireplaces, 0),  
 fireplace\_dum = as.numeric(n\_fireplaces > 0),  
 ADUdummy = as.numeric(ADUdummy > 0),  
 total\_canopy\_cov = replace\_na(total\_canopy\_cov, 0),  
 canopy\_dum = as.numeric(total\_canopy\_cov > 0),  
 sum\_z = zone\_test,  
 avg\_z = sum\_z/9,  
 z\_change = case\_when(avg\_z == 5 ~ 0,  
 TRUE ~ 1),  
 totalsqft\_sqd = totalsqft\*totalsqft,  
 taxlot\_area\_sqd = taxlot\_area\*taxlot\_area)  
  
#-------------------------------------------------------------------------  
  
# Clean SFR dataframe:   
# remove based on filtering parameters, set NAs as zero, add in improvements  
test2 <- test1 %>%  
 dplyr::select(-c(matches("zone"))) %>%  
 mutate\_at(vars(clean\_names), to0) %>%  
 mutate(arms\_length = price\_ratio > 20,  
 yearbuilt = na\_if(yearbuilt, 0),  
 saledate = mdy(saledate),   
 year\_sold = year(saledate),  
 age\_sold = year\_sold - yearbuilt,  
 percent\_vacant = percent\_vacant\*100,  
 SALEPRICElog = log(SALEPRICE)) %>%  
 filter(between(totalsqft, 1, 7500),   
 yearbuilt > 1500,  
 f\_baths < 6,  
 BLDGSQFT != 0,  
 age\_sold > 0,   
 top\_1 == FALSE,   
 arms\_length == TRUE,   
 vacant\_dummy == FALSE,   
 proud\_flag == FALSE,   
 llc\_flag == FALSE,  
 trust\_flag == FALSE)

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

#-------------------------------------------------------------------------  
  
# Select final relevant variables  
dat1 <- test2 %>%  
 dplyr::select(  
 STATE\_ID, RNO, PROPERTYID, TLID,   
 OWNER1, SITEADDR, SITEZIP, PRPCD\_DESC,   
 MKTVALYR3, LANDVAL3, BLDGVAL3, TOTALVAL3,   
 saledate, SALEPRICE, SALEPRICElog, age\_sold,  
 totalsqft, totalsqft\_sqd, taxlot\_area, taxlot\_area\_sqd, volume,   
 f\_baths, h\_baths, n\_fireplaces, fireplace\_dum, ADUdummy,   
 garage\_sqft, garage\_dum, deck\_sqft, deck\_dum,   
 attic\_sqft, attic\_dum, bsmt\_sqft, bsmt\_dum,   
 percent\_vacant, vacant\_dummy,  
 pct\_canopy\_cov, total\_canopy\_cov, canopy\_dum, zone\_change = z\_change,  
 CN\_score, Neighborhood = MapLabel, HIGH\_SCH, ELEM\_SCH, dist\_cityhall, dist\_ugb,  
 contains("con"), contains("pct")) %>%  
 dplyr::select(-c("CONCRETE", "FIN SECOND", "PAVING/CONCRETE ONLY", "UNF SECOND", "conFldway", "pct\_conFldway"))

# Summary Statistics

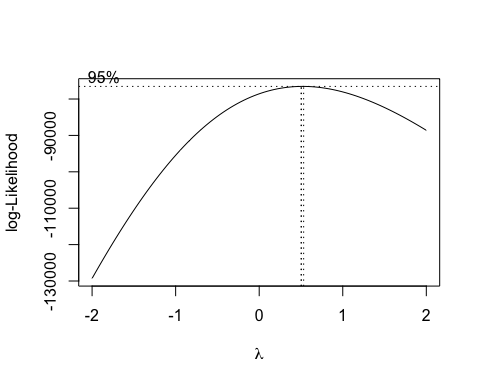
sumdat <- dat1 %>%  
 # Select and rename  
 dplyr::select(  
 `Assessed Land Value (2017)` = LANDVAL3,  
 `Assessed Market Value (2017)` = BLDGVAL3,  
 `Total Assessed Value (2017)` = TOTALVAL3,  
   
 `Sale Price` = SALEPRICE,  
 `Log Sale Price` = SALEPRICElog,  
 `Age When Sold` = age\_sold,  
 `Building Footprint (sqft)` = totalsqft,  
 `Building Footprint Squared (sqft)` = totalsqft\_sqd,  
 `Taxlot Area (sqft)` = taxlot\_area,  
 `Taxlot Area Squared (sqft)` = taxlot\_area\_sqd,  
 `Building Volume` = volume,  
   
 `Full Baths` = f\_baths,  
 `Half Baths` = h\_baths,  
 `Fireplaces` = n\_fireplaces,  
 `Fireplaces Dummy` = fireplace\_dum,  
 `Accessible Dwelling Unit Dummy` = ADUdummy,  
 `Garage Dummy` = garage\_dum,  
 `Garage Area (sqft)` = garage\_sqft,  
 `Attic Dummy` = attic\_dum,  
 `Attic Area (sqft)` = attic\_sqft,  
 `Basement Dummy` = bsmt\_dum,  
 `Basement Area (sqft)` = bsmt\_sqft,  
 `Deck Dummy` = deck\_dum,  
 `Deck Area (sqft)` = deck\_sqft,  
   
 `Vacant Properties in 200ft Radius (%)` = percent\_vacant,  
 `Vacant Properties in 200ft Radius Dummy` = vacant\_dummy,  
 `Canopy Cover (% of lot)` = pct\_canopy\_cov,  
 `Canopy Cover (sqft)` = total\_canopy\_cov,  
 `Canopy Cover Dummy` = canopy\_dum,  
 `Zoning Change Dummy` = zone\_change,  
 `Complete Neighborhoods Score (0-100)` = CN\_score,  
 `Distance to Central Business District (ft)` = dist\_cityhall,  
 `Distance to Urban Growth Boundary (ft)` = dist\_ugb,  
 #`Neighborhood Fixed Effects` = Neighborhood,  
 #`High School Fixed Effects` = HIGH\_SCH,  
 #`Elementary School Fixed Effects` = ELEM\_SCH,  
   
 `DEQ Environmental Cleanup Sites (ECSI)` = conECSI,  
 `DEQ Leaking Underground Storage Tank Cleanup Sites (LUST)` = conLUST,  
 `Historic and Conservation Districts` = conHist,  
 `Historic and Conservation Landmarks` = conECSI,  
 `Areas Requiring Archeological Scan or Consultation with Tribes` = conNatAm,  
 `Approach and Departure Cones` = conAirHgt,  
 `Helipad Landing` = conHeliprt,  
 `Airport Noise` = conNoise,  
 `Greenway` = conGW,  
 `DOGAMI Landslide Hazard Area` = conLSHA,  
 `DOGAMI Digital Landslide Database (SLIDO)` = conSLIDO,  
 `Slopes Over 25% Incline)` = conSlp25,  
 `Institutional Campuses` = conInstit,  
 `Private/Common Open Space` = conPrvCom,  
 `Publicly Owned Lots (non-residential)` = conPubOwn,  
 `Sewer System` = conSewer,  
 `Stormwater System` = conStorm,  
 `Water System` = conWater,  
 `Traffic Volume Exceeds Capacity` = conTranCap,  
 `ODOT Highway Interchanges` = conTranInt,  
 `Substandard and Unimproved Streets` = conTranSub,  
 `Scenic Views` = conView,  
 `Conservation Zones` = conCovrly,  
 `Preservation Zones` = conPovrly,  
 `Wetlands` = conWetland,  
 `FEMA 100-Year Floodplain Map` = conFld100,  
 `Percent of Preservation Zones` = pct\_conPovrly,  
 `Percent of Conservation Zones` = pct\_conCovrly,  
 `Percent of Wetlands` = pct\_conWetland,  
 `Percent of FEMA 100\_Year Floodplain Map` = pct\_conFld100,  
 )  
  
sumstat <- sumdat %>%  
 # Find the mean, st. dev., min, and max for each variable   
 summarise\_all(funs(mean, sd, min, max), na.rm = TRUE) %>%  
 pivot\_longer(everything(),  
 names\_to = c("key", "variable"),   
 names\_pattern = "(.\*)\_(.\*)") %>%  
 pivot\_wider(everything(),  
 names\_from = variable,  
 values\_from = value)  
  
# fix decimal places  
sumstat[,-1] <- round(sumstat[,-1], 2)  
  
  
  
# Write to .txt  
write.table(sumstat, file = "SFRsumstats.txt", sep = ",", quote = FALSE, row.names = F)

# Regression 1: Pre-analysis Formula and Functional Forms (Dummies)

model nomenclature Y\_Z Y = specifying feature Z = lin or log or bc .mod if regression results

# LINEAR REGRESSION-----------------------------------------  
  
# dummies only  
preanalysis\_lin = formula(paste0("SALEPRICE ~   
 age\_sold + totalsqft + totalsqft\_sqd + taxlot\_area +   
 taxlot\_area\_sqd + volume + f\_baths + h\_baths +   
 fireplace\_dum + ADUdummy +   
 garage\_dum + attic\_dum + bsmt\_dum + deck\_dum +  
 vacant\_dummy + canopy\_dum + zone\_change +  
 CN\_score + Neighborhood + HIGH\_SCH +   
 dist\_cityhall + dist\_ugb + ", con\_names))  
preanalysis\_lin.mod <- lm(preanalysis\_lin, dat1)  
  
# SEMI-LOG REGRESSION-------------------------------------------  
  
preanalysis\_log = formula(paste0("SALEPRICElog ~   
 age\_sold + totalsqft + totalsqft\_sqd + taxlot\_area +   
 taxlot\_area\_sqd + volume + f\_baths + h\_baths +   
 fireplace\_dum + ADUdummy +   
 garage\_dum + attic\_dum + bsmt\_dum + deck\_dum +  
 vacant\_dummy + canopy\_dum + zone\_change +  
 CN\_score + Neighborhood + HIGH\_SCH +   
 dist\_cityhall + dist\_ugb + ", con\_names))  
preanalysis\_log.mod <- lm(preanalysis\_log, dat1)  
  
# BOX-COX (LINEAR) REGRESSION--------------------------------------------  
  
bc = boxcox(preanalysis\_lin.mod, lamba = seq(-3, 3, 1))

## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :  
## extra argument 'lamba' will be disregarded



best\_lam = bc$x[which(bc$y == max(bc$y))]  
dat1 %<>%   
 mutate(SALEPRICEbc = SALEPRICE^best\_lam)  
  
preanalysis\_bc = formula(paste0("SALEPRICEbc ~   
 age\_sold + totalsqft + totalsqft\_sqd + taxlot\_area +   
 taxlot\_area\_sqd + volume + f\_baths + h\_baths +   
 fireplace\_dum + ADUdummy +   
 garage\_dum + attic\_dum + bsmt\_dum + deck\_dum +  
 vacant\_dummy + canopy\_dum + zone\_change +  
 CN\_score + Neighborhood + HIGH\_SCH +   
 dist\_cityhall + dist\_ugb + ", con\_names))  
preanalysis\_bc.mod <- lm(preanalysis\_bc, dat1)

preanal.star <- stargazer(  
 preanalysis\_lin.mod, preanalysis\_log.mod, preanalysis\_bc.mod,  
 title = "Regression #1: Pre-analysis Formula and Functional Forms (Dummy)",  
 column.labels = c("Linear", "Semi-Log", "Box-Cox")  
 )  
  
knitr::kable(preanal.star)  
# knitr::kable(preanal.star, "pandoc")  
# knitr::kable(head(dat1))

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Tue, Apr 21, 2020 - 14:34:12 \begin{table}[!htbp] \caption{Regression #1: Pre-analysis Formula and Functional Forms (Dummy)}

\end{table}