```
library(tidycensus)
library(tidyverse)
library(ggspatial)
library(dplyr)
library(ggplot2)
library(ggthemes)
library(fitdistrplus)
library(tibble)
library(skedastic)
library(lmtest)
library(BSDA)
library(sf)

census_api_key("74060ca8061c55486a9d98bfca2f006fb9772339",install=T)

# Q3
variables <- c(
'DP05_0001E','DP05_0018E','DP03_0062E','DP02_0065PE','DP03_0096PE','DP03_0128PE','DP04_004

data <- get_acs(geography = 'tract',
                state='IL',
                county='Cook County',
                output='wide',
                year=2019,
                geometry=TRUE,
                variables=variables,
                show_call=TRUE)

drop.cols <- grep("*M$", colnames(data))
data <- data[-(drop.cols)]

colnames(data) <- c(
'geoid','name','totpop','medage','medhhinc','propbac','propcov','proppov','proprent','geom

# Q4
ggplot(data) +
  geom_sf(aes(fill=propbac), alpha = 0.8) +
  annotation_scale(bar_cols = c("yellow2", "green")) +
  scale_fill_viridis_c(option='turbo') + theme_economist_white() +
  ggtitle(label = "AES 5-year (2015-2019) Estimates for Cook County, IL",
subtitle = "\nBaccalaureate Attainment Rates")

model <- lm(propbac ~ medhhinc, data)
summary(model)

# Q5
ggplot(data, aes(medhhinc,propbac)) +
  geom_point(color='red') +
  geom_smooth(method = 'lm') +
  theme_economist() +
  labs(x=substitute(paste(bold('Median Household Income ($)'))),
y=substitute(paste(bold('Baccalaureate Attainment Rates (%)'))), title = 'AES
5-year (2015-2019) Estimates for Cook County, IL') +
  theme(
```

```
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4)
  )

# Q6
mu_resd <- mean(model$residuals)
sd_resd <- sd(model$residuals)
len_resd <- length(model$residuals)

# ks-test check for normality
test <- ks.test(model$residuals, pnorm, mu_resd, sd_resd)
format(test$p.value, scientific=FALSE)

# plot to check normality of residuals
frame <- enframe(model$residuals)
x <- seq(-25, 25, length=len_resd)
ggplot(frame, aes(x=value)) +
  geom_histogram(fill='white', colour='black', alpha = 0.75, breaks =
seq(-30, 30, by = 2)) +
  stat_function(fun = function(x) dnorm(x, mean = mu_resd, sd = sd_resd) *
len_resd * 2, color='red', size=1) +
  theme_economist() +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5)
  ) +
  labs(x=substitute(paste(bold('Residual Value'))),
y=substitute(paste(bold('Count')))) +
  ggtitle(label = "Histogram for Residuals of the Model")

# test for serial correlation
dwtest(model)
# plot for serial correlation
par(mfrow = c(1,1))
df.residuals = data.frame(index(model$residuals), model$residuals)
colnames(df.residuals) <- c("Index", "Residuals")
ggplot(data=df.residuals, aes(x=Index, y=Residuals)) +
  geom_point(col='red', size=2) + ylim(-25, 25) + xlim(0, 1200) +
  theme_economist() + labs(x=substitute(paste(bold("Index"))),
y=substitute(paste(bold('Residuals')))) +
  ggtitle(label = "Plot for Serial Correlation of Residuals") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5)
  )

acf(model$residuals, type = "correlation", lag.max = 1)

# tests for heteroskedasticity
bptest(model)
#white_lm(model)
# plot for heteroskedasticity
```

```
ggplot(model, aes(y=model$residuals, x=na.omit(data$medhhinc))) +
  geom_point(col = 'red') + geom_abline(slope = 0) +
  theme_economist() +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5)
  ) +
  labs(x=substitute(paste(bold('Median Household Income'))),
y=substitute(paste(bold('Model Residuals')))) +
  ggtitle(label = "Plot for Heteroskedasticity")

# Q7
cor_og <- cor(data$propbac, data$medhhinc, use = "complete.obs")
samples <- replicate(10000, sample(data$medhhinc, nrow(data), replace=TRUE))
cors <- c()
for (i in 1:10000) {
  cors <- c(cors, cor(data$propbac, samples[,i], use = "complete.obs"))
}
significant_links <- cors[cors > cor_og]
length(significant_links) / length(cors)

# Q8
qplot(seq_along(cors), cors) +
  geom_point(color='red') +
  theme_economist() +
  labs(x=substitute(paste(bold(''))), y=substitute(paste(bold('')))) +
  ggtitle(label = "AES 5-year (2015-2019) Estimates for Cook County, IL",
subtitle = "\nCorrelation between 10k samples of Median Household Income and
Baccalaureate Attainment Rates") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4)
  )

ggplot(enframe(cors), aes(x=value)) +
  geom_histogram(color="black", fill="red") +
  theme_economist() +
  labs(x=substitute(paste(bold('Correlation'))),
y=substitute(paste(bold('Count')))) +
  ggtitle(label = "AES 5-year (2015-2019) Estimates for Cook County, IL",
subtitle = "\nCorrelation between 10k samples of Median Household Income and
Baccalaureate Attainment Rates") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4)
  )


# Q10
model.intercept <- model$coefficients[1]
model.slope <- model$coefficients[2]
slopes <- seq(-3 * model.slope, 5 * model.slope, 0.001 * model.slope)
```

```
sums_of_squares <- c()
sum_of_squares.get <- function(slope = model.slope, intercept =
model.intercept) {
  propbac.prediction <- intercept + slope * data$medhhinc
  sums_of_squares <<- c(sums_of_squares, sum((propbac.prediction -
data$propbac) ** 2, na.rm = TRUE))
}
mapply(sum_of_squares.get , slopes, model.intercept)

df.sse <- data.frame(slopes, sums_of_squares)
colnames(df.sse) <- c('Slope', 'SSE')

ggplot(df.sse,aes(x=Slope, y=SSE)) +
  geom_point(color='red', size = 1) +
  theme_economist() +
  labs(x=substitute(paste(bold('Slope'))), y=substitute(paste(bold('SSE'))))
+
  ggtitle(label = "Sum of Square of Residuals vs Slope", subtitle = "\nModel:
Baccalaureate Attainment Rates vs Median Household Income") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4)
  ) +
  annotate("point", x=0.0002577586, y=101495.9, color="Black") +
  annotate("text", x=0.0002577586, y=-150000, label="(0.0002577586,
101495.9)")

# Q11
intercepts <- seq(-8 * model.intercept, 10 * model.intercept, 0.001 *
model.intercept)
log_likelihood <- c()
log_likelihood.get <- function(slope = model.slope, intercept =
model.intercept) {
  propbac.prediction <- intercept + slope * data$medhhinc
  residuals <- data$propbac - propbac.prediction
  log_likelihood <<- c(log_likelihood, sum(dnorm(residuals, 0, summary(model)
$sigma, log=TRUE), na.rm = TRUE))
}

mapply(log_likelihood.get , model.slope, intercepts)

df.loglik <- data.frame(intercepts, log_likelihood)
colnames(df.loglik) <- c('Intercept', 'LogLikelihood')
ggplot(df.loglik,aes(x=Intercept, y=LogLikelihood)) +
  geom_point(color='red', size = 1) +
  theme_economist() +
  labs(x=substitute(paste(bold('Intercept'))), y=substitute(paste(bold('Log
Likelihood')))) +
  ggtitle(label = "Log Likelihood of Linear Model vs Intercept", subtitle =
"\nModel: Baccalaureate Attainment Rates vs Median Household Income") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4)
  ) +
```

```
  annotate("point", x=4.302643, y=-4720.427, color="Black") +
  annotate("text", x=4.302643, y=-4300, label="(4.302643, -4720.427)")



# Q12
medhhinc.robin_hood_policy <- sort(data$medhhinc, index.return=TRUE,
decreasing=TRUE)
len.medhhin <- length(medhhinc.robin_hood_policy$x)
medhhinc.robin_hood_policy$x[1:50] <- medhhinc.robin_hood_policy$x[1:50] -
10000
medhhinc.robin_hood_policy$x[-c(1:(len.medhhin-50))] <-
medhhinc.robin_hood_policy$x[-c(1:(len.medhhin-50))] + 10000

# Reversing the earlier sort to preserve the relationship between
# data points of propbac and medhhinc
medhhinc.robin_hood_policy <- data.frame(medhhinc.robin_hood_policy)
medhhinc.robin_hood_policy <-
medhhinc.robin_hood_policy[order(medhhinc.robin_hood_policy$ix), ]

propbac.robin_hood_prediction <- predict(model,
data.frame(medhhinc=medhhinc.robin_hood_policy$x))

# Plot to visualize the effect of Robin Hood Tax Policy on Baccalaureate
Attainment Rates
cols <- c("Before Robin Hood Tax Policy"="Blue","After Robin Hood Tax
Policy"="Red")
ggplot() +
  geom_point(aes(x=index(na.omit(data$propbac)), y=na.omit(data$propbac),
colour="Before Robin Hood Tax Policy")) +
  geom_point(aes(x=index(propbac.robin_hood_prediction),
y=propbac.robin_hood_prediction, colour="After Robin Hood Tax Policy")) +
  theme_economist() + labs(x=substitute(""),
y=substitute(paste(bold('Baccalaureate Attainment Rates')))) +
  ggtitle(label = "Effect of Robin Hood Policy on Baccalaureate Attainment
Rates") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5),
    legend.text.align = 1
  ) +
  scale_colour_manual(name="", values=cols)

cols <- c("Before Robin Hood Tax Policy"="Blue","After Robin Hood Tax
Policy"="Red")
ggplot() +
  geom_histogram(aes(x=na.omit(data$propbac), fill="Before Robin Hood Tax
Policy"), alpha=0.5) +
  geom_histogram(aes(x=propbac.robin_hood_prediction, fill="After Robin Hood
Tax Policy"), , alpha=0.5) +
  theme_economist() + labs(x=substitute(""),
y=substitute(paste(bold('Baccalaureate Attainment Rates')))) +
```

```
  ggtitle(label = "Effect of Robin Hood Policy on Baccalaureate Attainment
Rates") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5),
    legend.text.align = 1
  ) +
  scale_colour_manual(name="", values=cols, aesthetics = "fill")


# Theoretical Comparison of Baccalaureate Attainment Rates before and after
# Robin Hood Tax Policy
mean(data$propbac, na.rm=TRUE)
mean(propbac.robin_hood_prediction)

var(data$propbac, na.rm=TRUE)
var(propbac.robin_hood_prediction)



################################################################################

############# GROUP ASSIGNMENT 2 ############################

# Q2
model.v2 <- lm(propbac ~ totpop+medage+medhhinc+propcov+proppov+proprent,
data=data)

# a)
summary(model.v2)$r.squared
summary(model)$r.squared

# b)
anova(model, model.v2)

# c)
cols <- c("Old Model"="Blue","New Model"="Red")
df.residuals <- data.frame(model$residuals, model.v2$residuals)
colnames(df.residuals) <- c('OGModelResiduals', 'NewModelResiduals')
ggplot(df.residuals) +
  geom_density(aes(x=OGModelResiduals, fill='Old Model'), color='blue',
alpha=0.5) +
  geom_density(aes(x=NewModelResiduals, fill='New Model'), color='red',
alpha=0.5) +
  theme_economist() +
  labs(x=substitute(paste(bold("Residuals"))),
y=substitute(paste(bold('Density')))) +
  ggtitle(label = "Empirical Densities of Residuals from New and Old Model")
+
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5),
```

```
    legend.text.align = 10
  ) +
  scale_colour_manual(name="", values=cols, aesthetics = "fill")

df.residuals = data.frame(index(model.v2$residuals), model.v2$residuals)
colnames(df.residuals) <- c("Index", "Residuals")
ggplot(data=df.residuals, aes(x=Index, y=Residuals)) +
  geom_point(col='red', size=2) + ylim(-25, 25) + xlim(0, 1200) +
  theme_economist() + labs(x=substitute(paste(bold("Index"))),
y=substitute(paste(bold('Residuals')))) +
  ggtitle(label = "Plot for Serial Correlation of Residualsof New Model") +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5)
  )

acf(model.v2$residuals, type = "correlation", lag.max = 1)

# tests for heteroskedasticity
bptest(model.v2)
# plot for heteroskedasticity
ggplot(model, aes(y=model.v2$residuals, x=na.omit(model.v2$fitted.values))) +
  geom_point(col = 'red') + geom_abline(slope = 0) +
  theme_economist() +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5)
  ) +
  labs(x=substitute(paste(bold('New Model Fitted Values'))),
y=substitute(paste(bold('New Model Residuals')))) +
  ggtitle(label = "Plot for Heteroskedasticity")


# Q6
variables.v2 <- c('DP05_0001E','DP02_0065PE')

options(tigris_use_cache = TRUE)
college_education <- reduce(
  map(state.abb, function(x) {
    get_acs(geography = "tract", variables = variables.v2,
            state = x, output='wide', year=2019)
  }),
  rbind
)
drop.cols <- grep("*M$", colnames(college_education))
college_education <- college_education[-(drop.cols)]
colnames(college_education) <- c('geoid','name','totpop','propbac')
college_education$cook_county <- grepl('Cook County', college_education$name,
fixed = TRUE)

# a
college_education <- drop_na(college_education)
```

```
college_education <- college_education[college_education$totpop>=100,]

# b
mean.propbac <- mean(college_education[college_education$cook_county==FALSE,]
$propbac)
weighted.mean.propbac <-
weighted.mean(college_education[college_education$cook_county==FALSE,]
$propbac, college_education[college_education$cook_county==FALSE,]$totpop)

# c
#sd.propbac <- sd(college_education[college_education$cook_county==FALSE,]
$propbac)
#z.test(college_education[college_education$cook_county==TRUE,]$propbac,
mu=mean.propbac, sigma.x = sd.propbac, alternative='greater')
t.test(college_education[college_education$cook_county==TRUE,]$propbac,
mu=mean.propbac, alternative='greater')

# Q7
# a
require(sf)
nbc_tract <- '814.03'
prediction.data <- st_drop_geometry(subset(data, grepl(nbc_tract, name,
fixed=TRUE) == TRUE, select=c(-1,-2,-10)))
conf.int.pred.v2 <- predict(model.v2, newdata=prediction.data[-c(4)],
interval = "confidence", level=0.90)

# b
model.v3 <- lm(propbac ~ totpop+medage+medhhinc+propcov+proppov+proprent,
data=data, weights=data$totpop)
conf.int.pred.v3 <- predict(model.v3, prediction.data[-c(4)], interval =
"confidence", level=0.50)

# actual propbac for NBC Tract
prediction.data[4]

# c
rnormV <- Vectorize(rnorm)
coefficient.samples <- rnormV(10000, summary(model.v2)$coefficients[,1],
summary(model.v2)$coefficients[,2])
propbac.prediction.nbc_tract.10000 <- coefficient.samples %*%
c(1,data.matrix(prediction.data[-c(4)]))

mean.pred.nbc_tract <- mean(propbac.prediction.nbc_tract.10000)
sd.pred.nbc_tract <- sd(propbac.prediction.nbc_tract.10000)

conf.int.propbac.samples <- c(qnorm(0.05, mean.pred.nbc_tract,
sd.pred.nbc_tract),
                              qnorm(0.95, mean.pred.nbc_tract,
sd.pred.nbc_tract))


# Q8
ks.test(model.v3$residuals/summary(model.v3)$sigma, pnorm)
bptest(model.v3)
```

```
summary(model.v3)

data.v2 <- na.omit(data)
data.v2$residuals <- model.v3$residuals
fivenum(data.v2$residuals)[2]
data.v3 <- data.v2[data.v2$residuals <= fivenum(data.v2$residuals)[2], ]

ggplot(data.v3) +
  geom_sf(aes(fill=propbac), alpha = 0.8) +
  annotation_scale(bar_cols = c("yellow2", "green")) +
  scale_fill_viridis_c(option='turbo') + theme_economist_white() +
  ggtitle(label = "Areas in the lowest quartile of residuals", subtitle =
"\nBaccalaureate Attainment Rates")



data.v4 <- data[data$propbac <= fivenum(data$propbac)[2],]
ggplot(data.v4) +
  geom_sf(aes(fill=propbac), alpha = 0.8) +
  annotation_scale(bar_cols = c("yellow2", "green")) +
  scale_fill_viridis_c(option='turbo') + theme_economist_white() +
  ggtitle(label = "Areas in the lowest quartile of Baccalaureate Attainment
Rates")

# Q9
variables.v3 <-
c('DP03_0128PE','DP05_0018E','DP02_0065PE','DP02_0122PE','DP03_0096PE','DP03_0005P','DP05_

options(tigris_use_cache=TRUE)
#poverty_data <- reduce(
#  map(state.abb, function(x) {
#    get_acs(geography = "tract", variables = variables.v3,
#            state = x, geometry = TRUE, output='wide')
#  }),
#  rbind
#)
poverty_data <- get_acs(geography = 'tract',
                        state='IL',
                        county='Cook County',
                        output='wide',
                        year=2019,
                        geometry=TRUE,
                        variables=variables.v3,
                        show_call=TRUE)

drop.cols <- grep("*M$", colnames(poverty_data))
poverty_data <- poverty_data[-(drop.cols)]
colnames(poverty_data) <-
c('geoid','name','proppov','medage','propbac','propothlan','propcov','propunemp',
'propwhite','sexratio','hshldsz','estschool','propcomp', 'geometry')

#a
apply(poverty_data, 2, function(col) sum(is.na(col))/length(col))
poverty_data <- drop_na(poverty_data)
```

```
#b
basic.lm <-
lm(proppov~medage+propbac+propothlan+propcov+propunemp+propwhite+sexratio+hshldsz+estschoo
data=poverty_data)
forward <-
step(lm(proppov~1,poverty_data),scope=formula(basic.lm),direction='forward')

backward <- step(basic.lm,direction='backward')

#c
summary(basic.lm)$sigma
ks.test(basic.lm$residuals/summary(basic.lm)$sigma, pnorm)
bptest(basic.lm)
sqrt(mean(basic.lm$residuals^2))

# Q10
transformed.model <- lm(sqrt(proppov)~log(medage)+sqrt(propbac)+(propothlan)
+sqrt(propcov)+propunemp+propwhite+log(sexratio)+sqrt(hshldsz)
+sqrt(estschool)+sqrt(propcomp), data=poverty_data)

forward.transformed <-
step(lm(sqrt(proppov)~1,poverty_data),scope=formula(transformed.model),direction='forward'
backward.transformed <- step(transformed.model,direction='backward')

summary(transformed.model)$sigma
ks.test(transformed.model$residuals/summary(transformed.model)$sigma, pnorm)
bptest(transformed.model)
sqrt(mean(transformed.model$residuals^2))

mean.resd <- mean(transformed.model$residuals)
sd.resd <- sd(transformed.model$residuals)

ggplot(enframe(transformed.model$residuals), aes(x=value)) +
  geom_histogram(fill='white', colour='black', alpha = 0.75, breaks = seq(-5,
5, by = 0.5)) +
  stat_function(fun = function(x) dnorm(x, mean = mean.resd, sd = sd.resd) *
len_resd/2 , color='red', size=1) +
  theme_economist() +
  theme(
    axis.title.y = element_text(vjust = +4),
    axis.title.x = element_text(vjust = -4),
    plot.title = element_text(hjust = 0.5)
  ) +
  labs(x=substitute(paste(bold('Residual Value'))),
y=substitute(paste(bold('Count')))) +
  ggtitle(label = "Histogram for Residuals of the Model")
```