Group Project

Abrie Le Roux, Ali Raza, Jordan Keelan

2022-10-19

```r
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```r
library(tidyverse)
```

```
## ── Attaching packages
## ───────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✓ tibble  3.1.8     ✓ purrr   0.3.4
## ✓ tidyr   1.2.1     ✓ stringr 1.4.1
## ✓ readr   2.1.2     ✓ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ mosaic::count() masks dplyr::count()
## ✖ purrr::cross()  masks mosaic::cross()
## ✖ mosaic::do()    masks dplyr::do()
## ✖ tidyr::expand() masks Matrix::expand()
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ tidyr::pack()   masks Matrix::pack()
## ✖ mosaic::stat()  masks ggplot2::stat()
## ✖ mosaic::tally() masks dplyr::tally()
## ✖ tidyr::unpack() masks Matrix::unpack()
```

```r
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
shelter1.df = read.csv("https://open.alberta.ca/dataset/47f82be8-af8d-4994-8a97-2252d7643ff5/resource/b7080b66-25ea-4c30-ac4
7-02b64353637f/download/2013-2022-emergency-shelter-occupancy-machine-readable.csv")
```

Master Dataset

```
data = (shelter1.df %>% select(-City, -ShelterName, -Organization, -Shelter, -Capacity, -Daytime)) %>% na.omit() # remove un
needed columns
list(unique(data["ShelterType"]))
```

```
## [[1]]
##                          ShelterType
## 1                    Women Emergency
## 2                              Intox
## 3                    Adult Emergency
## 4                   Winter Emergency
## 5                    Youth Emergency
## 7             Short Term Supportive
## 31                  Family Emergency
## 81211          Long Term Supportive
## 96244      COVID19 Expanded Shelter
## 97797         COVID19 Isolation Site
## 97799 COVID19 Social Distancing Measures
```

```
data["Date"] <- as.Date(as.character(as.POSIXct(data$Date, format="%m/%d/%Y"))) # convert date column to date type
data = filter(data, ShelterType =='Adult Emergency'|ShelterType =='COVID19 Expanded Shelter'|ShelterType =='COVID19 social D
istancing Measures'|ShelterType =='Daytime Shelter'|ShelterType =='Family Emergency'|ShelterType =='Intox'|ShelterType =='Lo
ng Term Supportive'|ShelterType =='Short Term Supportive'|ShelterType =='Winter Emergency'|ShelterType =='Women Emergency'|S
helterType =='Youth Emergency')
data
```

Date <date>	ShelterType <chr>	Overnight <int>	YEAR <int>	MONTH <int>
2013-04-01	Women Emergency	65	2013	4
2013-04-01	Intox	74	2013	4
2013-04-01	Adult Emergency	253	2013	4
2013-04-01	Winter Emergency	152	2013	4
2013-04-01	Youth Emergency	51	2013	4
2013-04-01	Women Emergency	51	2013	4
2013-04-01	Short Term Supportive	143	2013	4
2013-04-01	Short Term Supportive	57	2013	4
2013-04-01	Short Term Supportive	21	2013	4
2013-04-01	Short Term Supportive	55	2013	4

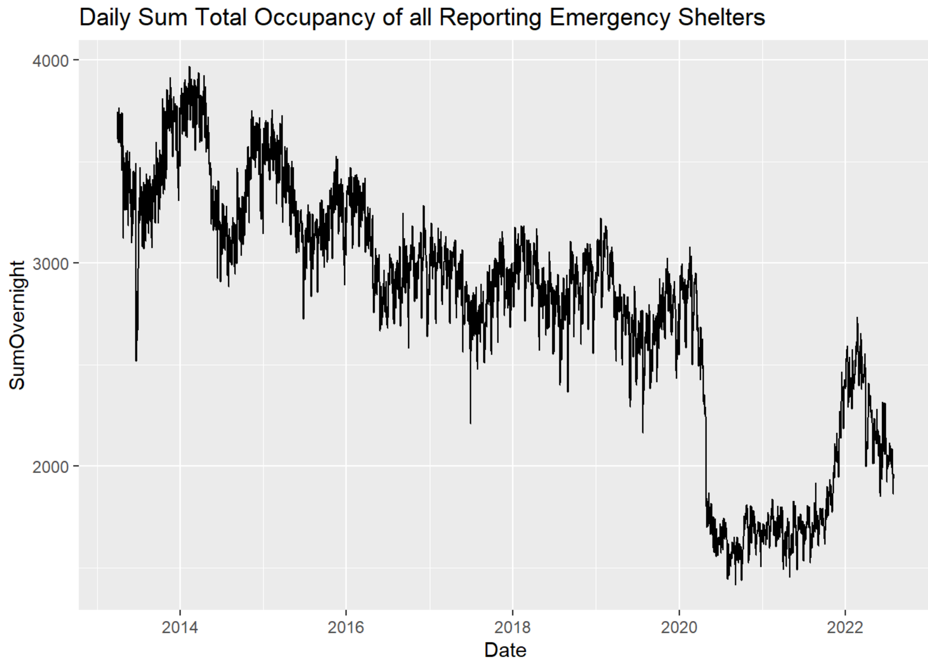
1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

Simplify by aggregating on daily total occupancy

```
daily.df <-(aggregate(data$Overnight, by=list(Date = data$Date),  FUN=sum)) # Sums total daily occupancy across all shelters
colnames(daily.df)[2] ="SumOvernight" # Renames new daily sum column
```

Daily sum total occupancy in all types of emergency shelters

```
ggplot(daily.df, aes(x=Date, y=SumOvernight)) +
  geom_line() +
  ggtitle("Daily Sum Total Occupancy of all Reporting Emergency Shelters")
```



Sum occupancy on quarter to be used in linear regression

```
data2 <- data # new data frame
data2$quarter <- as.yearqtr(data2$Date)  # Appends column that identifies quarter of entry
head(data2)
```

	Date <date>	ShelterType <chr>	Overnight <int>	YEAR <int>	MONTH <int>	quarter <yearqtr>
1	2013-04-01	Women Emergency	65	2013	4	2013 Q2
2	2013-04-01	Intox	74	2013	4	2013 Q2
3	2013-04-01	Adult Emergency	253	2013	4	2013 Q2
4	2013-04-01	Winter Emergency	152	2013	4	2013 Q2
5	2013-04-01	Youth Emergency	51	2013	4	2013 Q2
6	2013-04-01	Women Emergency	51	2013	4	2013 Q2

6 rows

```
quartely.df <-(aggregate(data2$Overnight, by=list(Quarter = data2$quarter),  FUN=sum)) # Sums total occupancy on quarter
colnames(quartely.df)[2] <- "SumOvernightQuarterly" # renames new summed column
quartely.df$index <-  seq.int(nrow(quartely.df)) # indexes data set
quartely.df <- filter(quartely.df, index > 3) # Removes quarters prior to 2014 Q1
quartely.df$index <-  seq.int(nrow(quartely.df)) # re-indexes data set
regression.df <- filter(quartely.df, index < 35) # Removed all quarters beyond 2019 Q4
regression.df # THIS IS OUR DATASET TO DO REGRESSION ON
```

Quarter <yearqtr>	SumOvernightQuarterly <int>	index <int>

Quarter <yearqtr>	SumOvernightQuarterly <int>	index <int>
2014 Q1	340986	1
2014 Q2	314549	2
2014 Q3	289877	3
2014 Q4	319089	4
2015 Q1	319330	5
2015 Q2	298349	6
2015 Q3	287737	7
2015 Q4	300116	8
2016 Q1	300610	9
2016 Q2	267818	10

1-10 of 34 rows Previous **1** 2 3 4 Next

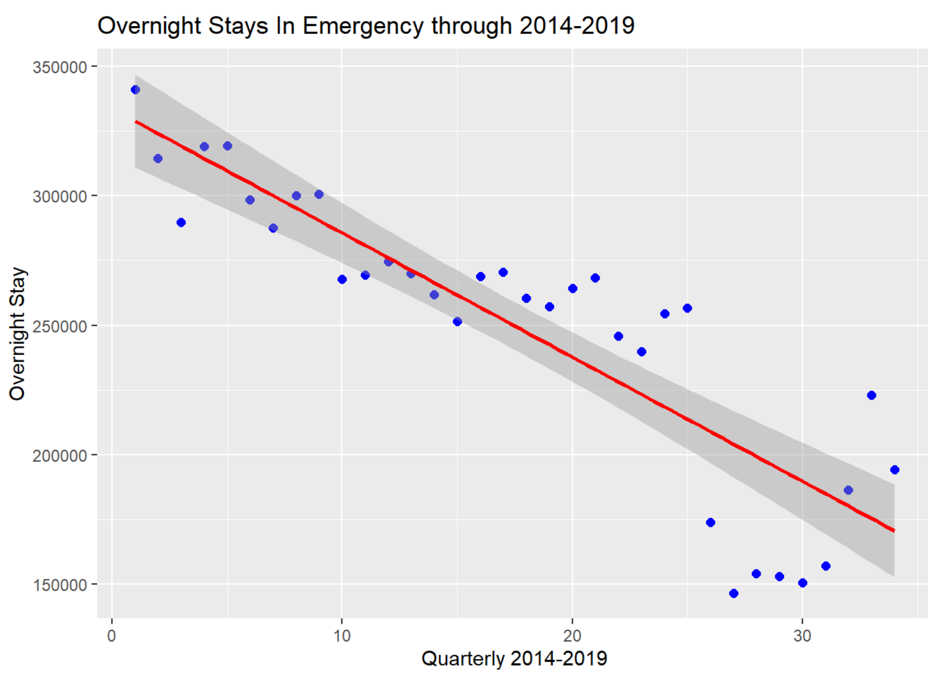
Ali Regression Code starts here

Below is the Linear Regression Model we are trying to prove

$$R_{SumOvernightQuarterly,i} = \beta_0 + \beta_1 * R_{Quarter,i} + e_i$$

```
ggplot(regression.df, aes(x = index, y = SumOvernightQuarterly)) + geom_point(col="blue", size = 2) + xlab("Quarterly 2014-2019") + ylab("Overnight Stay") + ggtitle("Overnight Stays In Emergency through 2014-2019") + geom_smooth(method="lm", col="red")
```

```
## `geom_smooth()` using formula 'y ~ x'
```



computing correlation coefficient

```
cor(~SumOvernightQuarterly, ~index, data=regression.df)
```

```
## [1] -0.8790499
```

$$r = -0.88323466989$$

Strong negative correlation..

Estimating the Model

```
predictovernight = lm(SumOvernightQuarterly ~ index, data=regression.df)
predictovernight$coef
```

```
## (Intercept)       index
##  333653.599   -4793.291
```

$\hat{R}_{SumOvernightQuarterly,i} = 2033.522347841533474 - 0.000059670087769 * \hat{R}_{Quarter,i}$ (Note: There is no e_i term on the estimate of the

#interpret the equation

Interpretation of b, estimate of B1:

As quarter decreases by 1 unit, then the for the occupancy rate will decrease by an *average* of -0.000059670087769.

Interpretation of b, estimate of B0: When the rate of the return of the market is 0 the rate of the overnight occupancy of shelters quarters stock is on average 2033.522347841533474.

```
summary(predictovernight)
```

```
##
## Call:
## lm(formula = SumOvernightQuarterly ~ index, data = regression.df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -57524 -12146   4709  15919  47543
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 333653.6     9219.4   36.19   <2e-16 ***
## index        -4793.3      459.5  -10.43    8e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26290 on 32 degrees of freedom
## Multiple R-squared:  0.7727, Adjusted R-squared:  0.7656
## F-statistic: 108.8 on 1 and 32 DF,  p-value: 7.998e-12
```

Squared is 0.77010819 which tells us that approx 77% of the variability observed can be explained by the regression model.

Below I am checking is the linearity of the model is valid #Should this be less than 0 since we are seeing a negative slope

$$\mathrm{H}_0 : \beta_1 = (\leq) 0 \qquad \mathrm{H}_A : \beta_1 < 0$$

computing our p value

```
coef(summary(predictovernight))
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 333653.599   9219.357  36.19055 1.550692e-27
## index        -4793.291    459.534 -10.43077 7.998163e-12
```

```
pt(-8.8344246814, 22)
```

```
## [1] 5.47167e-09
```

P value is less than 0.05 we reject null, therefore we can agree with our h alternative

Compute a 95% confidence interval for beta 1

```
qt(p = 0.025,df =57,lower.tail = FALSE )
```

```
## [1] 2.002465
```

```
-3268.4026087 - 369.9621341*(2.0024654593)
```

```
## [1] -4009.239
```

```
-3268.4026087 + 369.9621341*(2.0024654593)
```

```
## [1] -2527.566
```

$$-4009.2390035 <= B_1 <= -2527.5662139$$

normality of the residuals condition

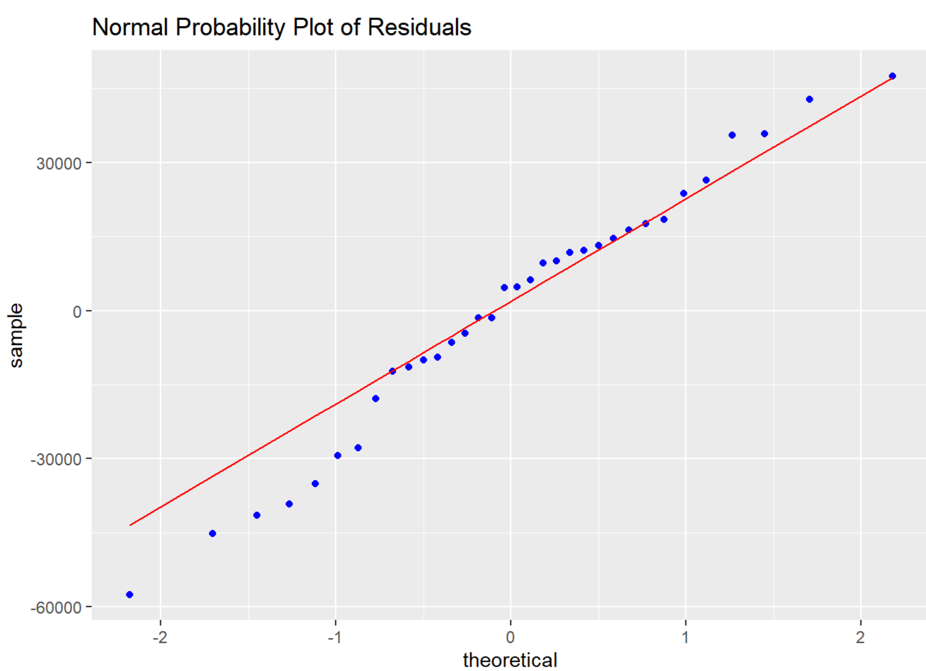
```
predicted.values.overnight = predictovernight$fitted.values #place the predicted values of y for each observed x into a vect
or
eison = predictovernight$residuals      #pull out the residuals
diagnosticdf2 = data.frame(predicted.values.overnight, eison) #create a data frame of fitted.values and residuals
```

```
diagnosticdf2
```

	predicted.values.overnight <dbl>	eison <dbl>
1	328860.3	12125.692
2	324067.0	-9518.016
3	319273.7	-29396.725
4	314480.4	4608.567
5	309687.1	9642.858
6	304893.9	-6544.851
7	300100.6	-12363.559
8	295307.3	4808.732
9	290514.0	10096.023
10	285720.7	-17902.685

1-10 of 34 rows Previous **1** 2 3 4 Next

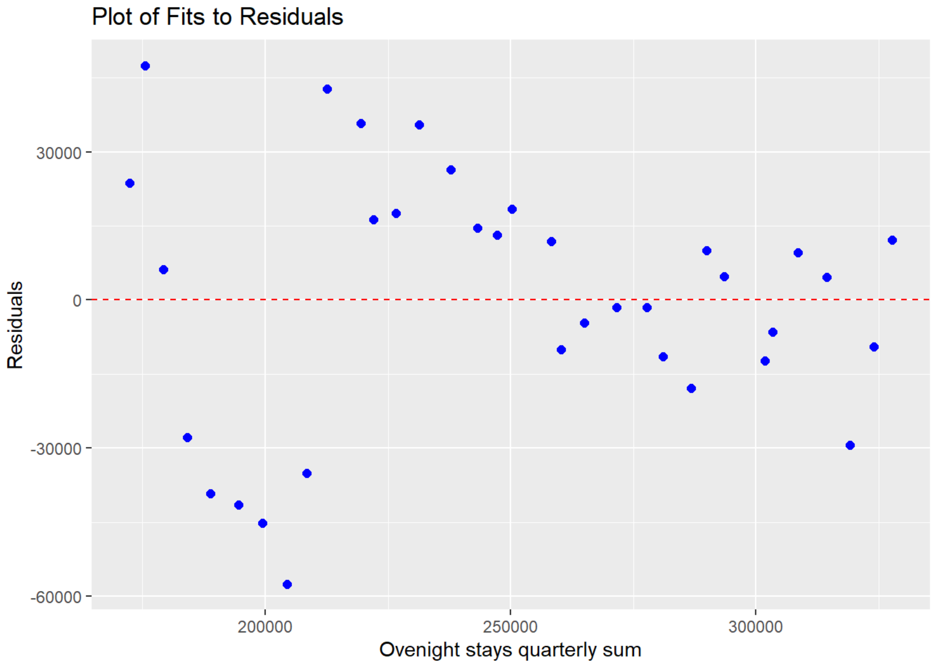
```
ggplot(diagnosticdf2, aes(sample = eison )) +  stat_qq(col='blue') + stat_qqline(col='red') + ggtitle("Normal Probability Pl
ot of Residuals")
```



residuals are normal

To inspect the homoscedasticity condition

```
ggplot(diagnosticdf2, aes(x = predicted.values.overnight, y = eison)) +  geom_point(size=2, col='blue', position="jitter") +
xlab("Ovenight stays quarterly sum") + ylab("Residuals") + ggtitle("Plot of Fits to Residuals") + geom_hline(yintercept=0, c
olor="red", linetype="dashed")
```




```
sum(diagnosticdf2$eison)
```

```
## [1] -9.094947e-12
```

really small 0 so we can say it is a good model when talking about the normality of residuals.

Below we will predict the number of overnight stays in emergency shelters 2020 q1 by using the predict function with index =25

```
predict(predictovernight, data.frame(index=31))
```

```
##        1
## 185061.6
```

```
predict(predictovernight, newdata=data.frame(index = 25), interval="conf") #compute the 95% CI for mean Y when x = 25
```

```
##        fit      lwr      upr
## 1 213821.3 202262.1 225380.6
```

95% confidence for the number of overnight stays in emergency shelters in the first quarter of 2020 will be between…

Below I am computing the r.boot,a.boot,b.boot,ymean.boot

```
Nbootstraps = 1000 #resample n =  200, 1000 times
cor.boot = numeric(Nbootstraps) #define a vector to be filled by the cor boot stat
a.boot = numeric(Nbootstraps) #define a vector to be filled by the a boot stat
b.boot = numeric(Nbootstraps) #define a vector to be filled by the b boot stat
ymean.boot = numeric(Nbootstraps) #define a vector to be filled by the predicted y boot stat
```

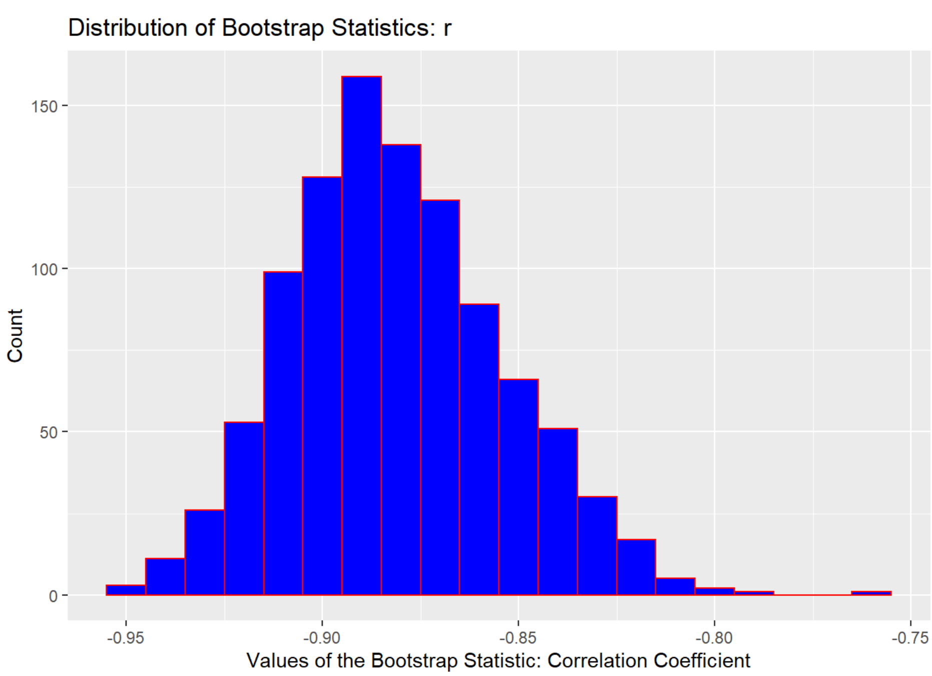
```
nsize = dim(regression.df)[1]  #set the n to be equal to the number of bivariate cases, number of rows
xvalue = 25 #set x = 15% for first quarter of 2020 in a certain county
#start of the for loop
for(i in 1:Nbootstraps)
{   #start of the loop
    index = sample(nsize, replace=TRUE)  #randomly picks a number between 1 and n, assigns as index
    demovote.boot = regression.df[index, ] #accesses the i-th row of the regression.df data frame
    #
    cor.boot[i] = cor(~SumOvernightQuarterly, ~index , data=demovote.boot) #computes correlation for each bootstrap sample
    votedemocrat.lm = lm(SumOvernightQuarterly ~ index, data=demovote.boot)  #set up the linear model
    a.boot[i] = coef(votedemocrat.lm)[1] #access the computed value of a, in position 1
    b.boot[i] = coef(votedemocrat.lm)[2] #access the computed value of b, in position 2
    ymean.boot[i] = a.boot[i] + (b.boot[i]*xvalue)
}
#end the loop
#create a data frame that holds the results of teach of he Nbootstraps
    bootstrapresultsdf = data.frame(cor.boot, a.boot, b.boot, ymean.boot)
```

```
bootstrapresultsdf
```

cor.boot <dbl>	a.boot <dbl>	b.boot <dbl>	ymean.boot <dbl>
-0.9021601	329470.4	-4993.819	204625.0
-0.9021272	342312.7	-5436.034	206411.9
-0.8737209	329071.9	-4458.081	217619.8
-0.8855634	336244.8	-4614.643	220878.7
-0.8339483	343839.1	-4917.620	220898.6
-0.8942921	340188.9	-5163.526	211100.7
-0.8702225	331678.5	-4773.617	212338.1
-0.8987139	340562.4	-5227.270	209880.7
-0.8862386	329816.7	-4159.038	225840.7
-0.8843109	334908.9	-5063.816	208313.5

1-10 of 1,000 rows Previous 1 2 3 4 5 6 ... 100 Next

```
ggplot(bootstrapresultsdf, aes(x = cor.boot)) + geom_histogram(col="red", fill="blue", binwidth=0.01) + xlab("Values of the
 Bootstrap Statistic: Correlation Coefficient") + ylab("Count") + ggtitle("Distribution of Bootstrap Statistics: r")
```



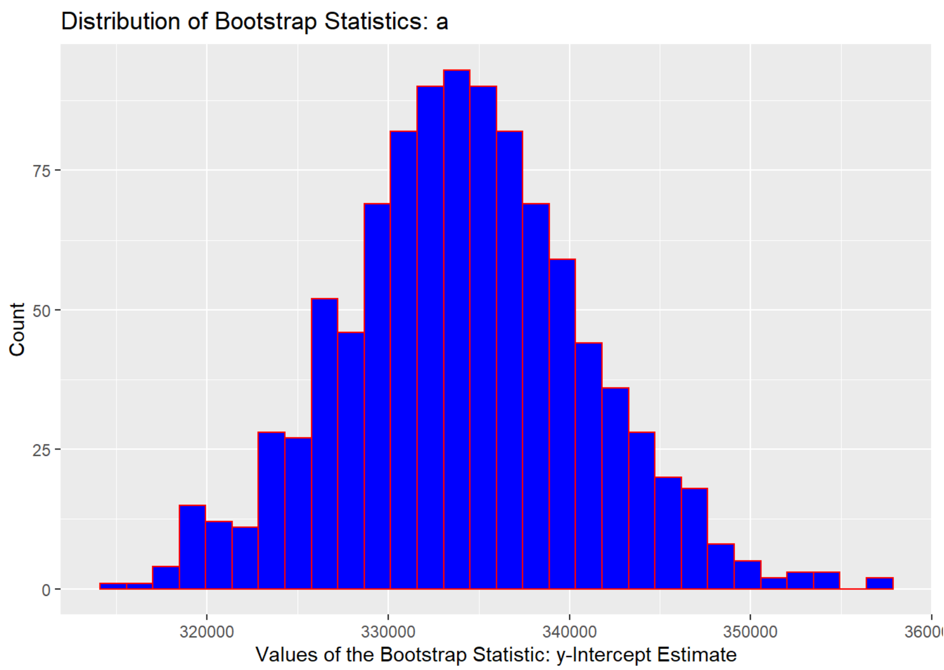
```
qdata(~cor.boot, c(0.025, 0.975), data=bootstrapresultsdf)
```

```
##       2.5%      97.5%
## -0.9303714 -0.8233200
```

$$-0.94398577475 <= r_{boot} <= -0.82093112577$$

```
ggplot(bootstrapresultsdf, aes(x = a.boot)) + geom_histogram(col="red", fill="blue") + xlab("Values of the Bootstrap Statist
ic: y-Intercept Estimate") + ylab("Count") + ggtitle("Distribution of Bootstrap Statistics: a")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



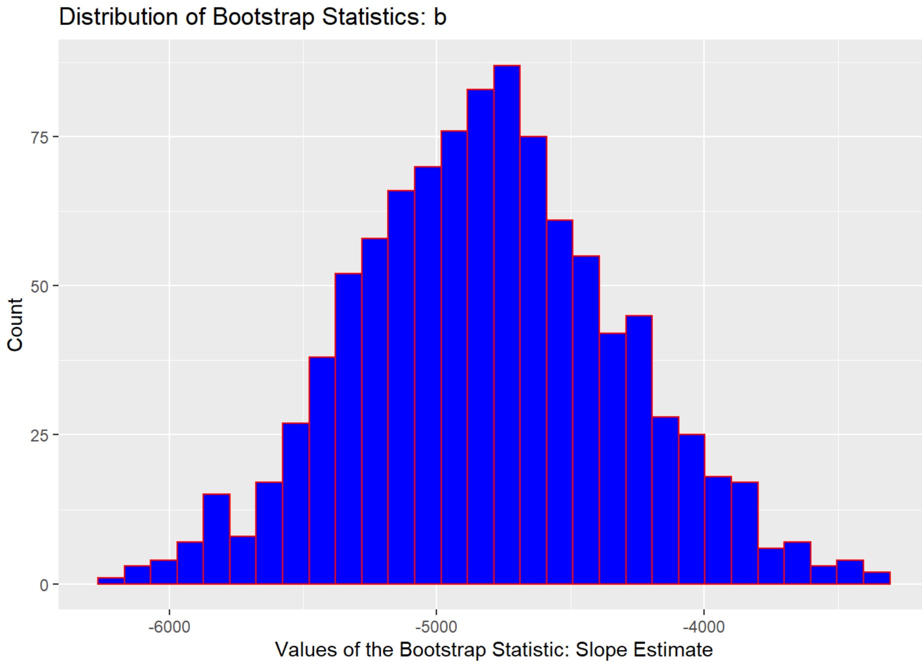

```
qdata(~a.boot, c(0.025, 0.975), data=bootstrapresultsdf)
```

```
##     2.5%    97.5%
## 320365.6 347203.2
```

$$306333.75162 <= a_{boot} <= 330773.08508$$

```
ggplot(bootstrapresultsdf, aes(x = b.boot)) + geom_histogram(col="red", fill="blue") + xlab("Values of the Bootstrap Statist
ic: Slope Estimate") + ylab("Count") + ggtitle("Distribution of Bootstrap Statistics: b")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



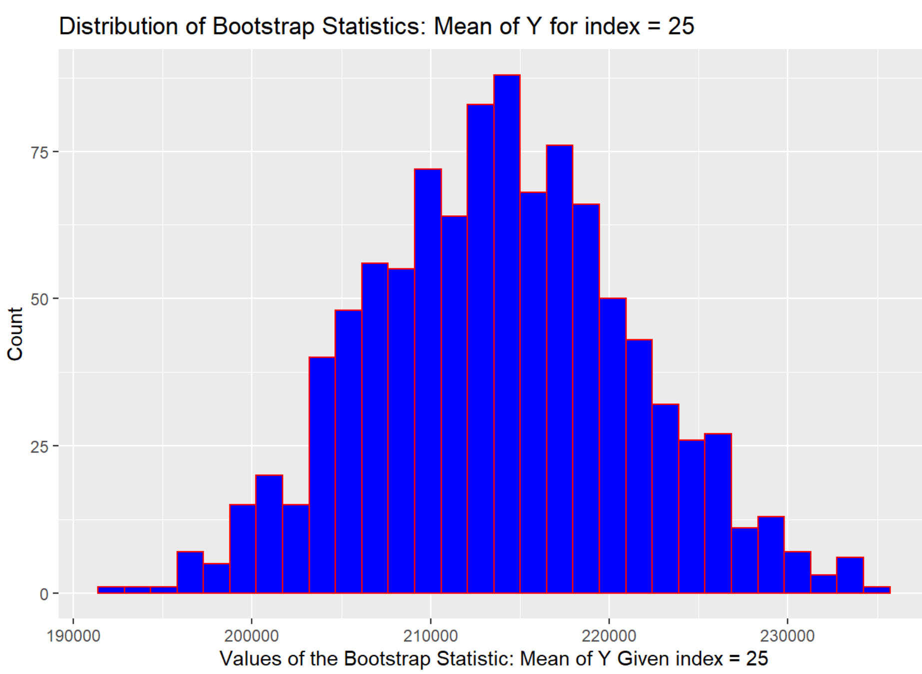

```
qdata(~b.boot, c(0.025, 0.975), data=bootstrapresultsdf)
```

```
##      2.5%     97.5%
## -5834.702 -3823.762
```

$$-4062.4420309 <= b_{boot} <= -2466.1849029$$

```
ggplot(bootstrapresultsdf, aes(x = ymean.boot)) + geom_histogram(col="red", fill="blue") + xlab("Values of the Bootstrap Sta
tistic: Mean of Y Given index = 25") + ylab("Count") + ggtitle("Distribution of Bootstrap Statistics: Mean of Y for index =
 25")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
qdata(~ymean.boot, c(0.025, 0.975), data=bootstrapresultsdf)
```

```
##     2.5%    97.5%
## 199977.8 228518.6
```

$$226560.25442 <= \mu_{y|x=25} <= 247034.23877$$

Set Data set up for women Shelters

```
data3 <- data

data3$Date <- floor_date(data3$Date, "month")

# Sum total occupants of womens shelters by month
womenData <- filter(data3, ShelterType=="Women Emergency")
data3.women <- aggregate(womenData$Overnight, by=list(Date=womenData$Date), FUN="sum")
colnames(data3.women)[2] <- "womenMonthOvernightSum"
data3.women
```

Date <date>	womenMonthOvernightSum <int>
2013-04-01	3584
2013-05-01	3408
2013-06-01	3210
2013-07-01	3401
2013-08-01	3514
2013-09-01	3334
2013-10-01	3220
2013-11-01	3371
2013-12-01	3441
2014-01-01	3846

1-10 of 112 rows Previous **1** 2 3 4 5 6 ... 12 Next

```
# Sum total occupants of all shelters by month
data3.all <- aggregate(data3$Overnight, by=list(Date=data3$Date), FUN="sum")
colnames(data3.all)[2] <- "totalMonthOvernightSum"
data3.all
```

Date <date>	totalMonthOvernightSum <int>
2013-04-01	107984
2013-05-01	105422
2013-06-01	94006
2013-07-01	101196
2013-08-01	101701
2013-09-01	101431
2013-10-01	109338
2013-11-01	111655
2013-12-01	112680
2014-01-01	116572

1-10 of 112 rows Previous **1** 2 3 4 5 6 ... 12 Next

```
# Combine Data Frames
data3.temp <- inner_join(data3.women,data3.all, by = "Date")
data3.temp$PropWomen <- data3.temp$womenMonthOvernightSum / data3.temp$totalMonthOvernightSum

# Remove Dates, splits data frame into one for each downturn
data3.downturn <- filter(filter(data3.temp, Date > "2014-09-01"), Date < "2016-10-01") #2014
data3.covid <- filter(filter(data3.temp, Date > "2020-03-01"), Date < "2022-04-01") #2020

# add indicator to each downturn
data3.downturn$Downturn = "2014-16"
data3.covid$Downturn = "2020-22"

# recombine
monthlywomen.df <- rbind(data3.downturn,data3.covid)
monthlywomen.df
```

Date	womenMonthOvernightSum	totalMonthOvernightSum	PropWomen	Downturn
<date>	<int>	<int>	<dbl>	<chr>
2014-10-01	3504	103121	0.03397950	2014-16
2014-11-01	3651	107224	0.03405021	2014-16
2014-12-01	3785	108744	0.03480652	2014-16
2015-01-01	4060	110720	0.03666908	2014-16
2015-02-01	3626	99567	0.03641769	2014-16
2015-03-01	3939	109043	0.03612336	2014-16
2015-04-01	3644	102618	0.03551034	2014-16
2015-05-01	3567	101802	0.03503860	2014-16
2015-06-01	3249	93929	0.03458996	2014-16
2015-07-01	3380	96643	0.03497408	2014-16

1-10 of 48 rows Previous **1** 2 3 4 5 Next

```
#keep dates for prop chart
```

```
favstats(~ PropWomen | Downturn, data=monthlywomen.df)
```

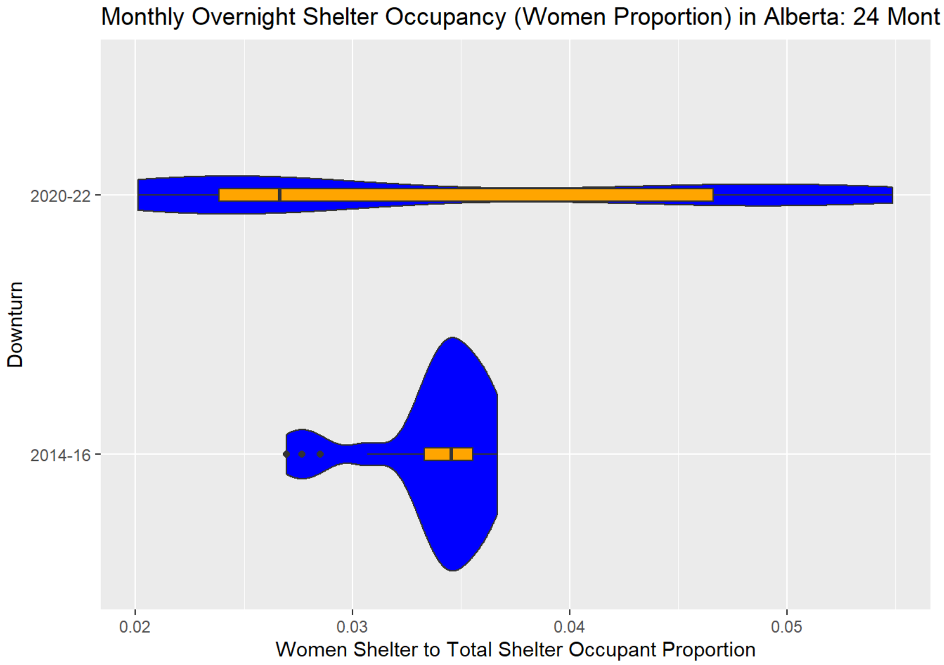
Downturn	min	Q1	median	Q3	max	mean	sd	n	missing
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<int>	<int>
2014-16	0.02695498	0.03329420	0.03451880	0.03554781	0.03666908	0.03375491	0.002700619	24	0
2020-22	0.02013068	0.02384516	0.02662194	0.04661333	0.05487817	0.03410784	0.012489177	24	0

2 rows

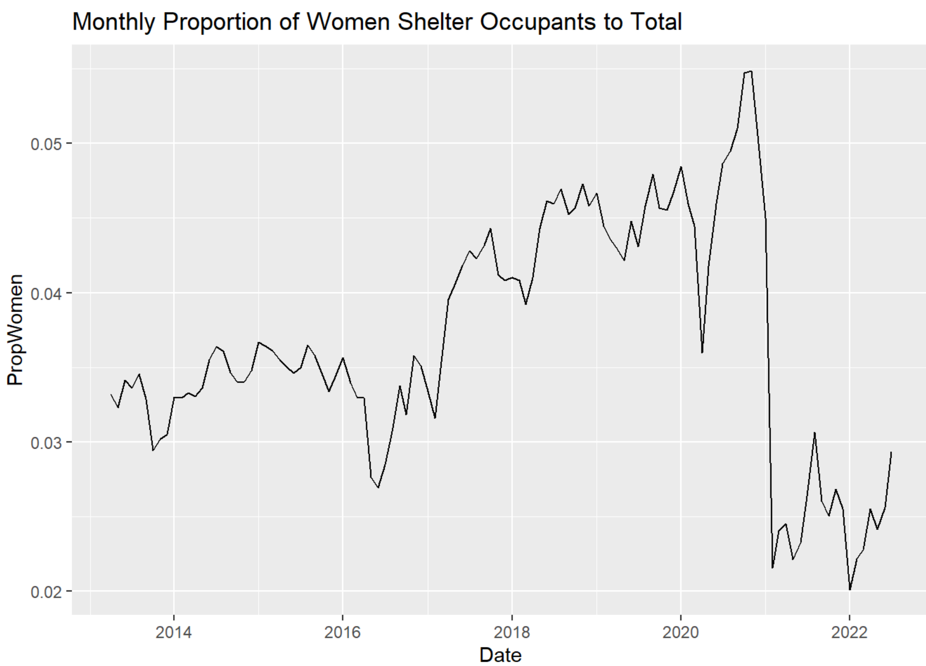
```
0.03666908-0.03329420
```

```
## [1] 0.00337488
```

```
ggplot(data=monthlywomen.df, aes(x = Downturn, y = PropWomen)) + geom_violin(fill="blue") + geom_boxplot(width = 0.05, fill=
"orange") + xlab("Downturn") + ylab("Women Shelter to Total Shelter Occupant Proportion") + ggtitle("Monthly Overnight Shelt
er Occupancy (Women Proportion) in Alberta: 24 Month") + coord_flip()
```




```
ggplot(data3.temp, aes(x=Date, y=PropWomen)) +
  geom_line() +
  ggtitle("Monthly Proportion of Women Shelter Occupants to Total")
```



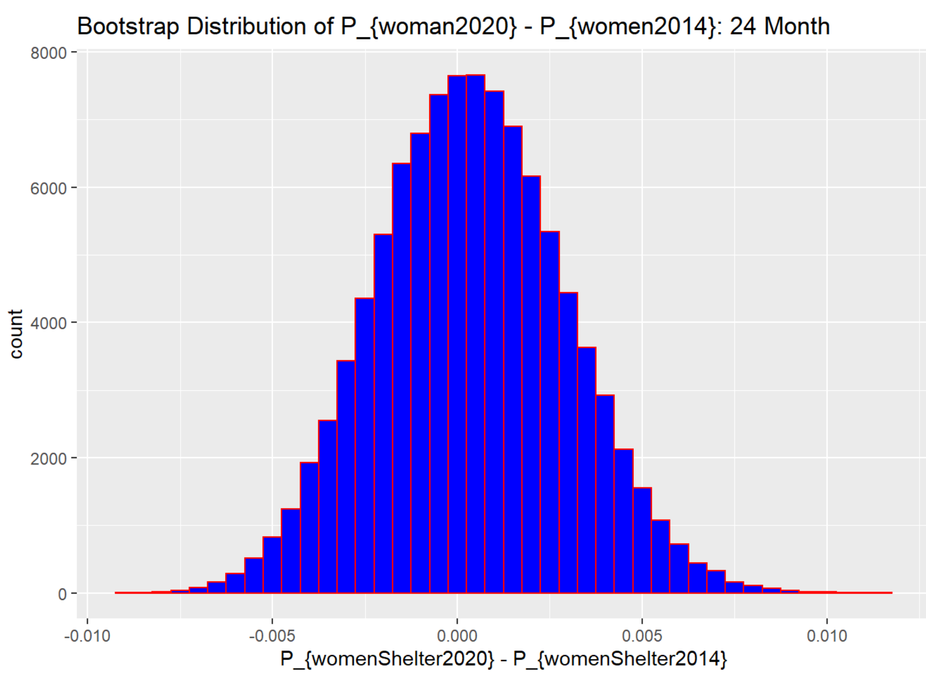

```
n.2014 = favstats(~totalMonthOvernightSum|Downturn, data=monthlywomen.df)$n[1]
n.2020 = favstats(~totalMonthOvernightSum|Downturn, data=monthlywomen.df)$n[2]
NsimsW = 100000
prop.2014 = numeric(NsimsW)
prop.2020 = numeric(NsimsW)
diff.props = numeric(NsimsW)

data.2014w = filter(monthlywomen.df, Downturn=="2014-16")
data.2020w = filter(monthlywomen.df, Downturn=="2020-22")
```

```
for(i in 1:NsimsW)
  {   prop.2014[i] = mean(sample(data.2014w$PropWomen, n.2014, replace=TRUE))
      prop.2020[i] = mean(sample(data.2020w$PropWomen, n.2020, replace=TRUE))
      diff.props[i] = prop.2020[i] - prop.2014[i]
}

boot.women = data.frame(prop.2020, prop.2014, diff.props)
head(boot.women,100)
```

	prop.2020 <dbl>	prop.2014 <dbl>	diff.props <dbl>
1	0.03316998	0.03407852	-9.085393e-04
2	0.03614281	0.03351692	2.625884e-03
3	0.03277605	0.03422084	-1.444784e-03
4	0.03155652	0.03300916	-1.452637e-03
5	0.03605052	0.03361129	2.439232e-03
6	0.02751051	0.03400034	-6.489836e-03
7	0.03052253	0.03402383	-3.501306e-03
8	0.03735754	0.03360433	3.753202e-03
9	0.03501343	0.03366138	1.352045e-03
10	0.03365686	0.03354108	1.157823e-04

1-10 of 100 rows Previous **1** 2 3 4 5 6 ... 10 Next

```
ggplot(data=boot.women, aes(x = diff.props)) + geom_histogram(fill='blue', col='red', binwidth=.0005) + xlab("P_{womenShelter2020} - P_{womenShelter2014}") + ggtitle("Bootstrap Distribution of P_{woman2020} - P_{women2014}: 24 Month")
```




```
qdata(~ diff.props, c(0.025, 0.975), data=boot.women)
```

```
##         2.5%        97.5%
## -0.004483631  0.005453888
```

$$95\%CI: -0.00452 < p_{womanShelter2020} - p_{womanShelter2014} < 0.0055$$

Ali Permutation test code starts here on the prop of women proportion difference

```
favstats(~ totalMonthOvernightSum | Downturn, data=monthlywomen.df)
```

Downturn <chr>	min <dbl>	Q1 <dbl>	median <dbl>	Q3 <dbl>	max <dbl>	mean <dbl>	sd <dbl>	n <int>	missing <int>
2014-16	84400	93766.25	99446.5	102727.5	110720	98436.75	6911.490	24	0
2020-22	47140	50896.00	52167.0	57183.5	76434	56058.88	9317.918	24	0

2 rows

```
favstats(~ totalMonthOvernightSum | Downturn, data=monthlywomen.df)[1,]$mean - favstats(~ totalMonthOvernightSum | Downturn, data=monthlywomen.df)[2,]$mean
```

```
## [1] 42377.88
```

```
favstats(~ PropWomen | Downturn, data=monthlywomen.df)
```

Downturn <chr>	min <dbl>	Q1 <dbl>	median <dbl>	Q3 <dbl>	max <dbl>	mean <dbl>	sd <dbl>	n <int>	missing <int>
2014-16	0.02695498	0.03329420	0.03451880	0.03554781	0.03666908	0.03375491	0.002700619	24	0
2020-22	0.02013068	0.02384516	0.02662194	0.04661333	0.05487817	0.03410784	0.012489177	24	0

2 rows

```
favstats(~ PropWomen | Downturn, data=monthlywomen.df)[1,]$mean - favstats(~ PropWomen | Downturn, data=monthlywomen.df)[2,]$mean
```

```
## [1] -0.0003529306
```

```
obMeanDiff = favstats(~ PropWomen | Downturn, data=monthlywomen.df)[2,]$mean -
  favstats(~ PropWomen | Downturn, data=monthlywomen.df)[1,]$mean #computes current difference of sample means
obMeanDiff
```

```
## [1] 0.0003529306
```

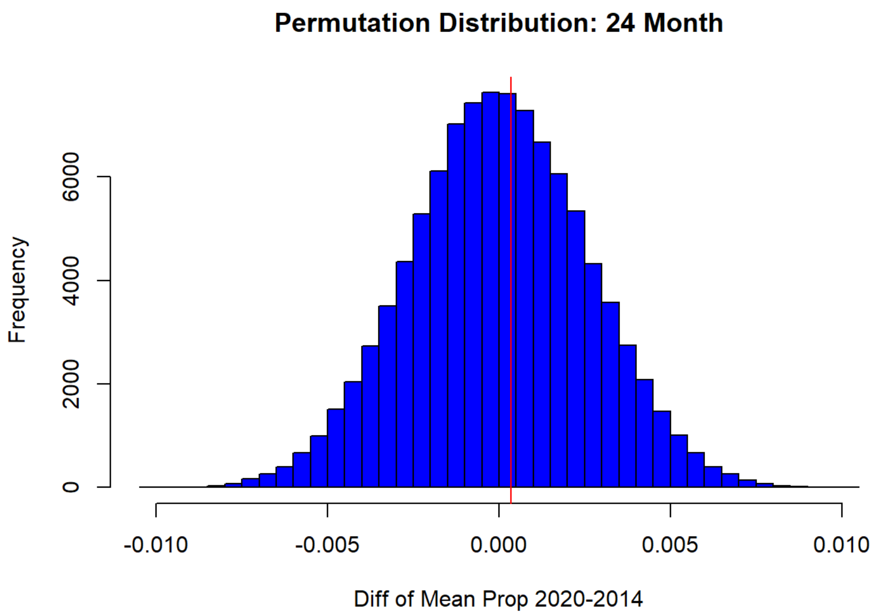
```
N = 100000 #2000 different permutations minus the difference we have observed
womenprop.2014=numeric(N)
womenprop.2020=numeric(N)
outcomeW = numeric(N) #create a vector to store differences of means
for(i in 1:N)
{ indexW = sample(48, 24, replace=FALSE)
   womenprop.2014[i] = mean(monthlywomen.df$PropWomen[indexW])
   womenprop.2020[i] = mean(monthlywomen.df$PropWomen[-indexW])
   outcomeW[i] = womenprop.2020[i] - womenprop.2014[i] #difference between means
}

diffWomen.df.12=data.frame(womenprop.2020,womenprop.2014,outcomeW)
diffWomen.df.12
```

womenprop.2020 <dbl>	womenprop.2014 <dbl>	outcomeW <dbl>
0.03413542	0.03372733	4.080905e-04
0.03439439	0.03346836	9.260283e-04
0.03506156	0.03280119	2.260374e-03
0.03467248	0.03319027	1.482213e-03
0.03416447	0.03369828	4.661960e-04
0.03288298	0.03497977	-2.096790e-03
0.03421281	0.03364994	5.628647e-04
0.03122568	0.03663707	-5.411389e-03
0.03177120	0.03609155	-4.320344e-03

womenprop.2020	womenprop.2014	outcomeW
<dbl>	<dbl>	<dbl>
0.03472698	0.03313577	1.591214e-03

1-10 of 10,000 rows Previous 1 2 3 4 5 6 ... 1000 Next

```
hist(outcomeW, xlab="Diff of Mean Prop 2020-2014", ylab="Frequency", main="Permutation Distribution: 24 Month", col='blue',
 breaks=50)
abline(v = obMeanDiff, col="red")
```




```
p.value = prop(outcomeW >= obMeanDiff)
p.value
```

```
## prop_TRUE
##   0.44329
```

12 month test

```
# Remove Dates, splits data frame into one for each downturn
data4.downturn <- filter(filter(data3.temp, Date > "2014-09-01"), Date < "2015-10-01") #2014
data4.covid <- filter(filter(data3.temp, Date > "2020-03-01"), Date < "2021-04-01") #2020

# add indicator to each downturn
data4.downturn$Downturn = "2014-2015"
data4.covid$Downturn = "2020-2021"

# recombine
monthly12women.df <- rbind(data4.downturn,data4.covid)
monthly12women.df
```

Date	womenMonthOvernightSum	totalMonthOvernightSum	PropWomen	Downturn
<date>	<int>	<int>	<dbl>	<chr>
2014-10-01	3504	103121	0.03397950	2014-2015
2014-11-01	3651	107224	0.03405021	2014-2015
2014-12-01	3785	108744	0.03480652	2014-2015
2015-01-01	4060	110720	0.03666908	2014-2015
2015-02-01	3626	99567	0.03641769	2014-2015
2015-03-01	3939	109043	0.03612336	2014-2015

Date <date>	womenMonthOvernightSum <int>	totalMonthOvernightSum <int>	PropWomen <dbl>	Downturn <chr>
2015-04-01	3644	102618	0.03551034	2014-2015
2015-05-01	3567	101802	0.03503860	2014-2015
2015-06-01	3249	93929	0.03458996	2014-2015
2015-07-01	3380	96643	0.03497408	2014-2015

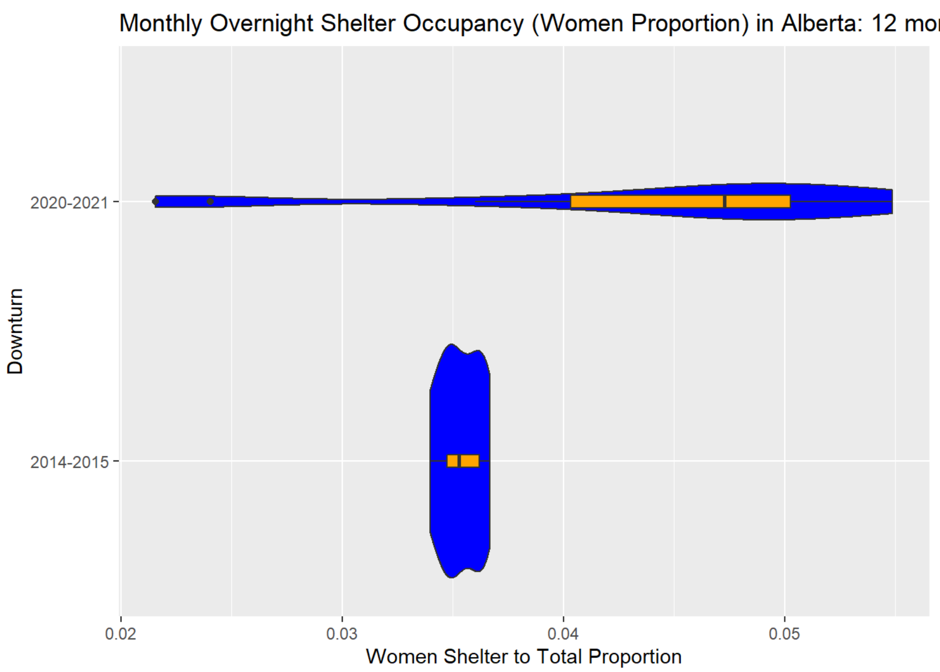
1-10 of 24 rows Previous **1** 2 3 Next

```
favstats(~ PropWomen | Downturn, data=monthly12women.df)
```

Downturn <chr>	min <dbl>	Q1 <dbl>	median <dbl>	Q3 <dbl>	max <dbl>	mean <dbl>	sd <dbl>	n <int>
2014-2015	0.03397950	0.03475238	0.03527447	0.03619695	0.03666908	0.03537220	0.0009391506	12
2020-2021	0.02156514	0.04033757	0.04728875	0.05025456	0.05487817	0.04359259	0.0110506349	12

2 rows | 1-9 of 10 columns

```
ggplot(data=monthly12women.df, aes(x = Downturn, y = PropWomen)) + geom_violin(fill="blue") + geom_boxplot(width = 0.05, fil
l="orange") + xlab("Downturn") + ylab("Women Shelter to Total Proportion") + ggtitle("Monthly Overnight Shelter Occupancy (W
omen Proportion) in Alberta: 12 month") + coord_flip()
```



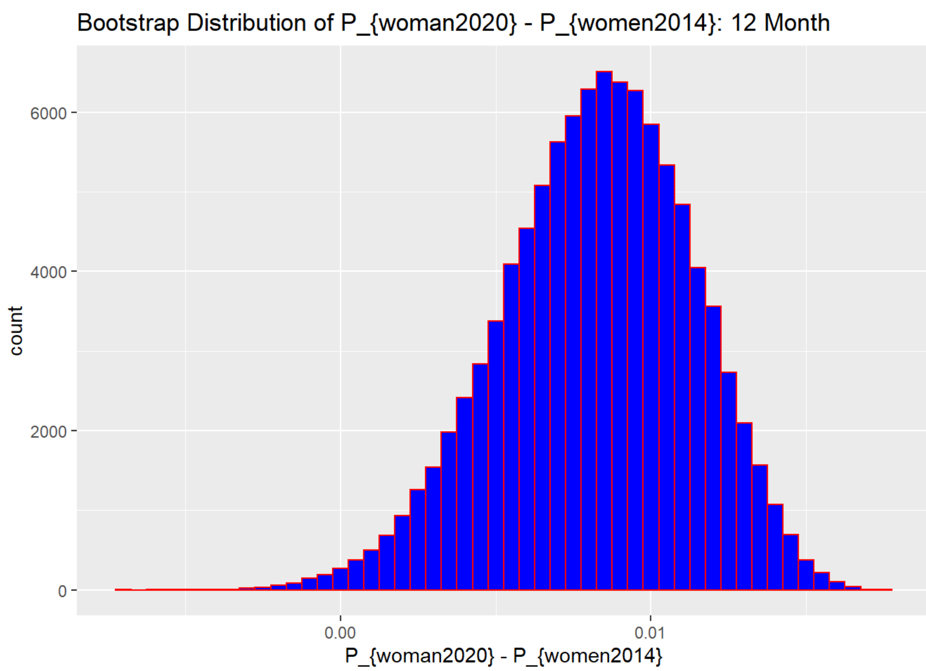

```
n.2014.12 = favstats(~totalMonthOvernightSum|Downturn, data=monthly12women.df)$n[1]
n.2020.12 = favstats(~totalMonthOvernightSum|Downturn, data=monthly12women.df)$n[2]
NsimsW = 100000
prop.12.2014 = numeric(NsimsW)
prop.12.2020 = numeric(NsimsW)
diff.props.12 = numeric(NsimsW)

data.2014.12 = filter(monthly12women.df, Downturn=="2014-2015")
data.2020.12 = filter(monthly12women.df, Downturn=="2020-2021")
```

```
for(i in 1:NsimsW)
  {   prop.12.2014[i] = mean(sample(data.2014.12$PropWomen, n.2014.12, replace=TRUE))
      prop.12.2020[i] = mean(sample(data.2020.12$PropWomen, n.2020.12, replace=TRUE))
      diff.props.12[i] = prop.12.2020[i] - prop.12.2014[i]
}

boot.women.12 = data.frame(prop.12.2020, prop.12.2014, diff.props.12)
head(boot.women.12,100)
```

	prop.12.2020 <dbl>	prop.12.2014 <dbl>	diff.props.12 <dbl>
1	0.04145027	0.03542064	0.0060296223
2	0.05069791	0.03525190	0.0154460063
3	0.03857398	0.03582774	0.0027462367
4	0.04524136	0.03505631	0.0101850482
5	0.04831944	0.03550066	0.0128187810
6	0.03978704	0.03544690	0.0043401467
7	0.04710368	0.03587725	0.0112264241
8	0.05049849	0.03544001	0.0150584856
9	0.04656586	0.03556172	0.0110041427
10	0.04558954	0.03558394	0.0100056003

1-10 of 100 rows Previous 1 2 3 4 5 6 ... 10 Next

```
ggplot(data=boot.women.12, aes(x = diff.props.12)) + geom_histogram(fill='blue', col='red', binwidth=.0005) + xlab("P_{woman2020} - P_{women2014}") + ggtitle("Bootstrap Distribution of P_{woman2020} - P_{women2014}: 12 Month")
```




```
qdata(~ diff.props.12, c(0.025, 0.975), data=boot.women.12)
```

```
##        2.5%       97.5%
## 0.001821467 0.013749273
```

Ali Permutation test code starts here on the prop of women proportion difference

```
favstats(~ totalMonthOvernightSum | Downturn, data=monthly12women.df)
```

Downturn <chr>	min <dbl>	Q1 <dbl>	median <dbl>	Q3 <dbl>	max <dbl>	mean <dbl>	sd <dbl>	n <int>	missing <int>
2014-2015	93929	96471.5	102210	107604.0	110720	102042.08	5904.136	12	0
2020-2021	47140	48688.0	51197	52520.5	71343	52340.75	6351.352	12	0

2 rows

```
favstats(~ totalMonthOvernightSum | Downturn, data=monthly12women.df)[1,]$mean - favstats(~ totalMonthOvernightSum | Downturn, data=monthly12women.df)[2,]$mean
```

```
## [1] 49701.33
```

```
favstats(~ PropWomen | Downturn, data=monthly12women.df)
```

Downturn <chr>	min <dbl>	Q1 <dbl>	median <dbl>	Q3 <dbl>	max <dbl>	mean <dbl>	sd <dbl>	n <int>
2014-2015	0.03397950	0.03475238	0.03527447	0.03619695	0.03666908	0.03537220	0.0009391506	12
2020-2021	0.02156514	0.04033757	0.04728875	0.05025456	0.05487817	0.04359259	0.0110506349	12

2 rows | 1-9 of 10 columns

```
favstats(~ PropWomen | Downturn, data=monthly12women.df)[1,]$mean - favstats(~ PropWomen | Downturn, data=monthly12women.df)[2,]$mean
```

```
## [1] -0.008220392
```

```
obMeanDiff.12 = favstats(~ PropWomen | Downturn, data=monthly12women.df)[2,]$mean -
  favstats(~ PropWomen | Downturn, data=monthly12women.df)[1,]$mean #computes current difference of sample means
obMeanDiff.12
```

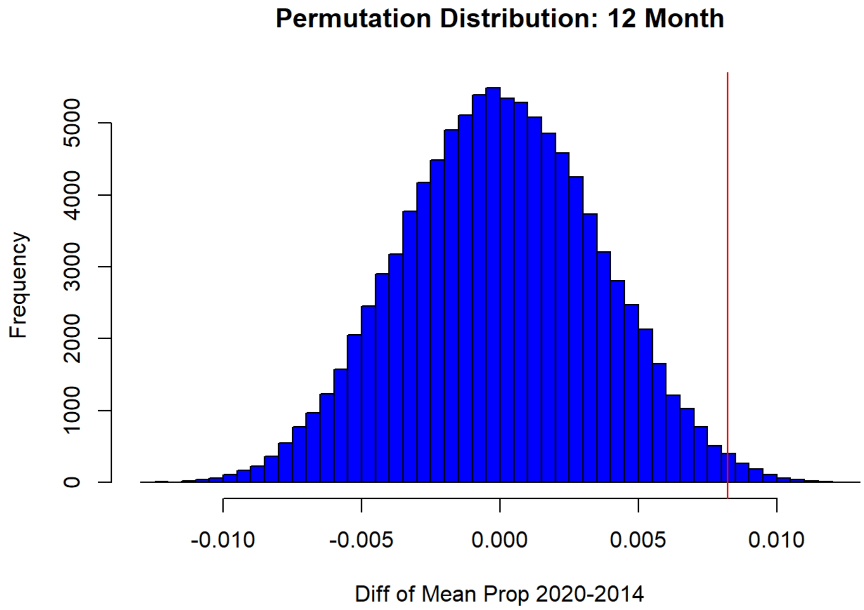
```
## [1] 0.008220392
```

```
N = 100000 #2000 different permutations minus the difference we have observed
womenprop.2014.12=numeric(N)
womenprop.2020.12=numeric(N)
outcomeW.12 = numeric(N) #create a vector to store differences of means
for(i in 1:N)
{ indexW.12 = sample(24, 12, replace=FALSE)
   womenprop.2014.12[i] = mean(monthly12women.df$PropWomen[indexW.12])
   womenprop.2020.12[i] = mean(monthly12women.df$PropWomen[-indexW.12])
   outcomeW.12[i] = womenprop.2020.12[i] - womenprop.2014.12[i] #difference between means
}

diffWomen.df.12=data.frame(womenprop.2020.12,womenprop.2014.12,outcomeW.12)
diffWomen.df.12
```

womenprop.2020.12 <dbl>	womenprop.2014.12 <dbl>	outcomeW.12 <dbl>
0.04049832	0.03846647	2.031850e-03
0.03853317	0.04043161	-1.898438e-03
0.03818821	0.04077657	-2.588361e-03
0.03945678	0.03950800	-5.122378e-05
0.04041250	0.03855229	1.860208e-03
0.04395100	0.03501379	8.937208e-03
0.03991554	0.03904925	8.662947e-04
0.03862889	0.04033590	-1.707014e-03
0.04076573	0.03819906	2.566674e-03
0.04032451	0.03864028	1.684229e-03

1-10 of 10,000 rows Previous **1** 2 3 4 5 6 ... 1000 Next

```
hist(outcomeW.12, xlab="Diff of Mean Prop 2020-2014", ylab="Frequency", main="Permutation Distribution: 12 Month", col='blue', breaks=50)
abline(v = obMeanDiff.12, col="red")
```




```
p.value.12 = prop(outcomeW.12 >= obMeanDiff.12)
p.value.12
```

```
## prop_TRUE
##   0.00896
```