



One-Stop Analytics: Predictive Modeling (Linear Models)

Case Study of Autism Spectrum Disorder (ASD) with R



ABOUT 1 IN 59 CHILDREN

WERE IDENTIFIED WITH AUTISM SPECTRUM DISORDER
AMONG A 2014 SAMPLE OF 8 YEAR OLDS FROM 11 US COMMUNITIES
IN CDC'S ADDM NETWORK

[United States]

Centers for Disease Control and Prevention (CDC) - Autism Spectrum Disorder (ASD)

Autism spectrum disorder (ASD) is a developmental disability that can cause significant social, communication and behavioral challenges. CDC is committed to continuing to provide essential data on ASD, search for factors that put children at risk for ASD and possible causes, and develop resources that help identify children with ASD as early as possible.

<https://www.cdc.gov/ncbdd/autism/data/index.html> (<https://www.cdc.gov/ncbdd/autism/data/index.html>)

[Singapore]

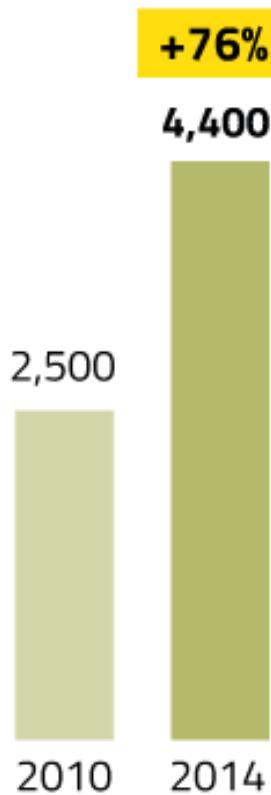
TODAY Online - More preschoolers diagnosed with developmental issues

Doctors cited better awareness among parents and preschool teachers, leading to early referrals for diagnosis.

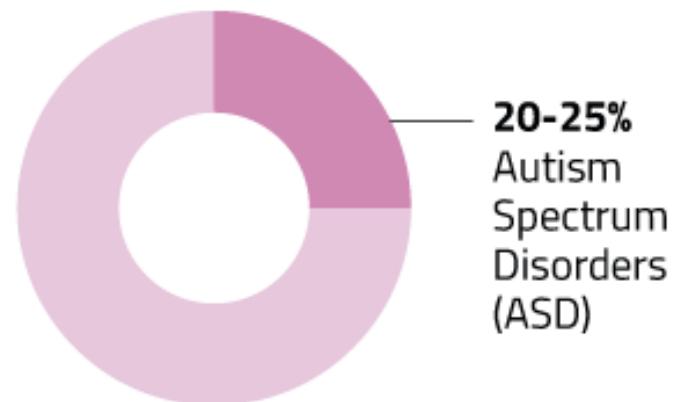
<https://www.gov.sg/news/content/today-online-more-preschoolers-diagnosed-with-developmental-issues>
<https://www.gov.sg/news/content/today-online-more-preschoolers-diagnosed-with-developmental-issues>

Jump in preschoolers diagnosed with developmental issues

● New cases



● Types of diagnosed cases



● ASD incidence

i / 160

World Health
Organisation

i / 150

Singapore

Source: KK Women's and Children's Hospital, National University Hospital TODAY

 PATHLIGHT SCHOOL
Where lives are transformed

[Home](#) [About Us](#) [Programmes](#) [Admissions](#) [Happenings](#) [Support Us](#) [Careers](#) [News](#)

1ST AUTISM-FOCUSED SCHOOL that offers a unique blend of mainstream academics & life readiness skills

 Highlights
Latest events and happenings at Pathlight School.

 The Art Faculty
Support the products by individuals with autism.

 e-Learning Portals
» Learn for Life eCampus
» MC Online
» Student Learning Space

 Parents' Corner
Useful resources and information for our parents

<https://www.pathlight.org.sg/> (<https://www.pathlight.org.sg/>)

.0

Workshop Objective:

Use R to predict Autism Spectrum Disorder (ASD) prevalence.

<https://www.cdc.gov/ncbdd/autism/data/index.html> (<https://www.cdc.gov/ncbdd/autism/data/index.html>)

- **Linear Model: Simple Linear Regression (SLR)**
- **Linear Model: Multiple Linear Regression (MLR)**
- **Linear Model: Polynomial Regression (PLR)**
- **Linear Model: Logistic Regression (LR)**
- **Linear Model: Model Evaluation: Train/Test, K-Fold Cross Validation, Confusion Matrix**
- **Workshop Submission**
- **Appendices**

.0

```
In [1]: library("repr") # Show graphs in-line notebook
```

Obtain current R working directory

```
In [2]: getwd()
```

'/media/sf_vm_shared_folder/git/DDC/DDC-ASD/model_R'

Set new R working directory

```
In [3]: # setwd("/media/sf_vm_shared_folder/git/DDC/DDC-ASD/model_R")
# setwd('~/Desktop/admin-desktop/vm_shared_folder/git/DDC-ASD/model_R')
getwd()
```

'/media/sf_vm_shared_folder/git/DDC/DDC-ASD/model_R'

.0

Linear Model: Simple Linear Regression (SLR)

Linear Model: Simple Linear Regression (SLR) - Workshop Task

Workshop Task:

1. a. Graph the data in a scatterplot to determine if there is a possible linear relationship.
2. b. Compute and interpret the linear correlation coefficient, r .
3. c. Determine the regression equation for the data.
4. d. Graph the regression equation and the data points.
5. e. Identify potential influential observations (outliers).
6. f. At the 5% significance level, do the data provide sufficient evidence to conclude that the slope of the population regression line is not 0 and, hence, that [Year] is useful as a predictor of ASD [Prevalence]?
7. g. Obtain the residuals and create a residual plot. Decide whether it is reasonable to consider that the assumptions for regression analysis are met by the variables in questions.
8. h. Compute and interpret the coefficient of determination, R^2 .
9. i. Find the predicted ASD Prevalence of future Year.
10. j. Determine a 95% confidence interval for the predicted ASD Prevalence.

Use Case Data: ["../dataset/ADV_ASD_State_R.csv"](#)

Read in CSV data, storing as R dataframe

```
In [4]: # Read back in above saved file:  
ASD_State <- read.csv("../dataset/ADV_ASD_State_R.csv")  
ASD_State$Year_Factor <- factor(ASD_State$Year_Factor, ordered = TRUE) # Conve  
ASD_State$Prevalence_Risk2 = factor(ASD_State$Prevalence_Risk2, ordered=TRUE,  
ASD_State$Prevalence_Risk4 = factor(ASD_State$Prevalence_Risk4, ordered=TRUE,
```

```
In [5]: head(ASD_State)
```

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_F
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizo
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Geo
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	Marylan

```
In [6]: # Filter [ Source: ADDM ], including only two columns for SLR:  
# Dependent variable: Prevalence  
# independent variable: Year  
ASD_State_4_SLR = subset(ASD_State, Source_UC == 'ADDM', select = c(Prevalence  
#  
dim(ASD_State_4_SLR)  
head(ASD_State_4_SLR)
```

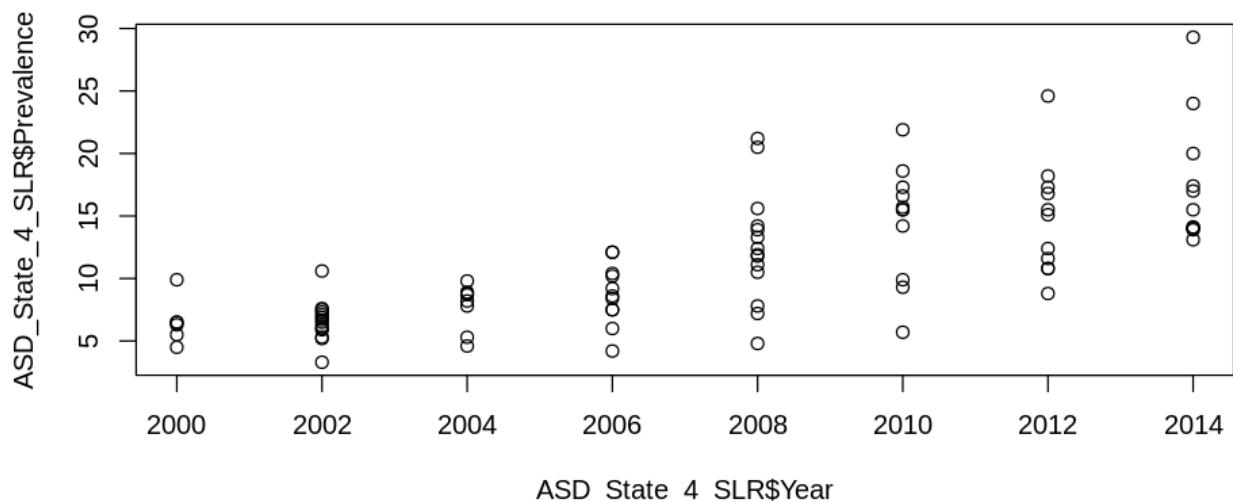
86 2

Prevalence	Year
6.5	2000
6.5	2000
5.5	2000
9.9	2000
6.3	2000
4.5	2000

SLR Workshop Task: 1. a. Graph the data in a scatterplot to determine if there is a possible linear relationship.

```
In [7]: # Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=4)
```

```
In [8]: plot(ASD_State_4_SLR$Year, ASD_State_4_SLR$Prevalence)
```



SLR Workshop Task: 2. b. Compute and interpret the linear correlation coefficient, r.

Compute correlation coefficient

```
In [9]: cor(ASD_State_4_SLR$Year, ASD_State_4_SLR$Prevalence)
```

0.722409821134655

Apply correlation test (two tail: != 0)

```
In [10]: cor.test(ASD_State_4_SLR$Year, ASD_State_4_SLR$Prevalence)
```

Pearson's product-moment correlation

```
data: ASD_State_4_SLR$Year and ASD_State_4_SLR$Prevalence
t = 9.5753, df = 84, p-value = 4.13e-15
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.6027995 0.8102653
sample estimates:
cor
0.7224098
```

Apply correlation test (one tail: > 0)

```
In [11]: cor.test(ASD_State_4_SLR$Year, ASD_State_4_SLR$Prevalence, alternative = "grea
```

Pearson's product-moment correlation

```
data: ASD_State_4_SLR$Year and ASD_State_4_SLR$Prevalence
t = 9.5753, df = 84, p-value = 2.065e-15
alternative hypothesis: true correlation is greater than 0
95 percent confidence interval:
0.6243611 1.0000000
sample estimates:
cor
0.7224098
```

SLR Workshop Task: 3. c. Determine the regression equation for the data.

```
In [12]: fit_model = lm(formula = Prevalence ~ Year, data = ASD_State_4_SLR)
print(fit_model)
```

Call:

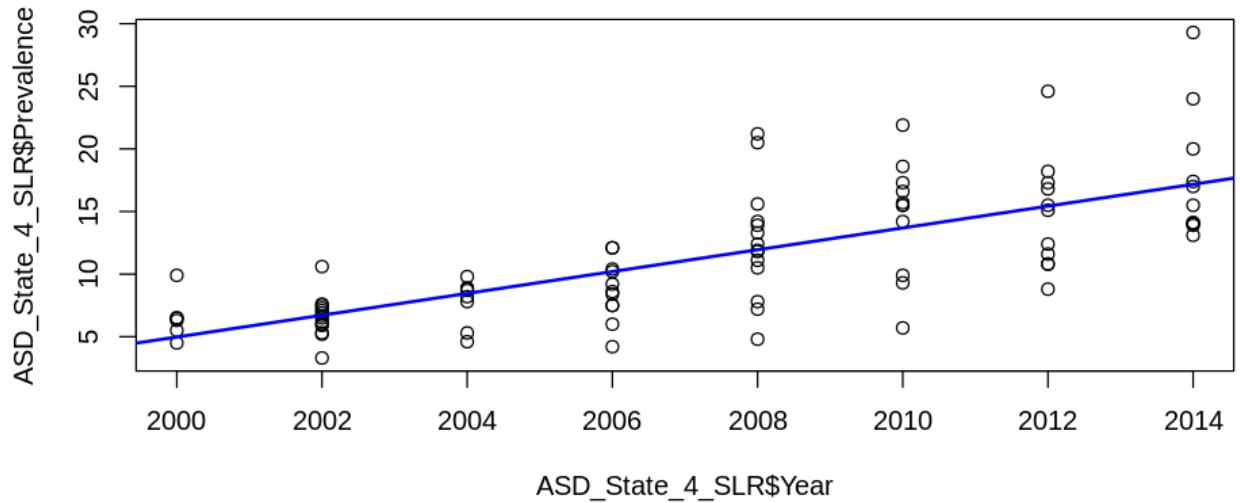
lm(formula = Prevalence ~ Year, data = ASD_State_4_SLR)

Coefficients:

(Intercept)	Year
-1737.8464	0.8714

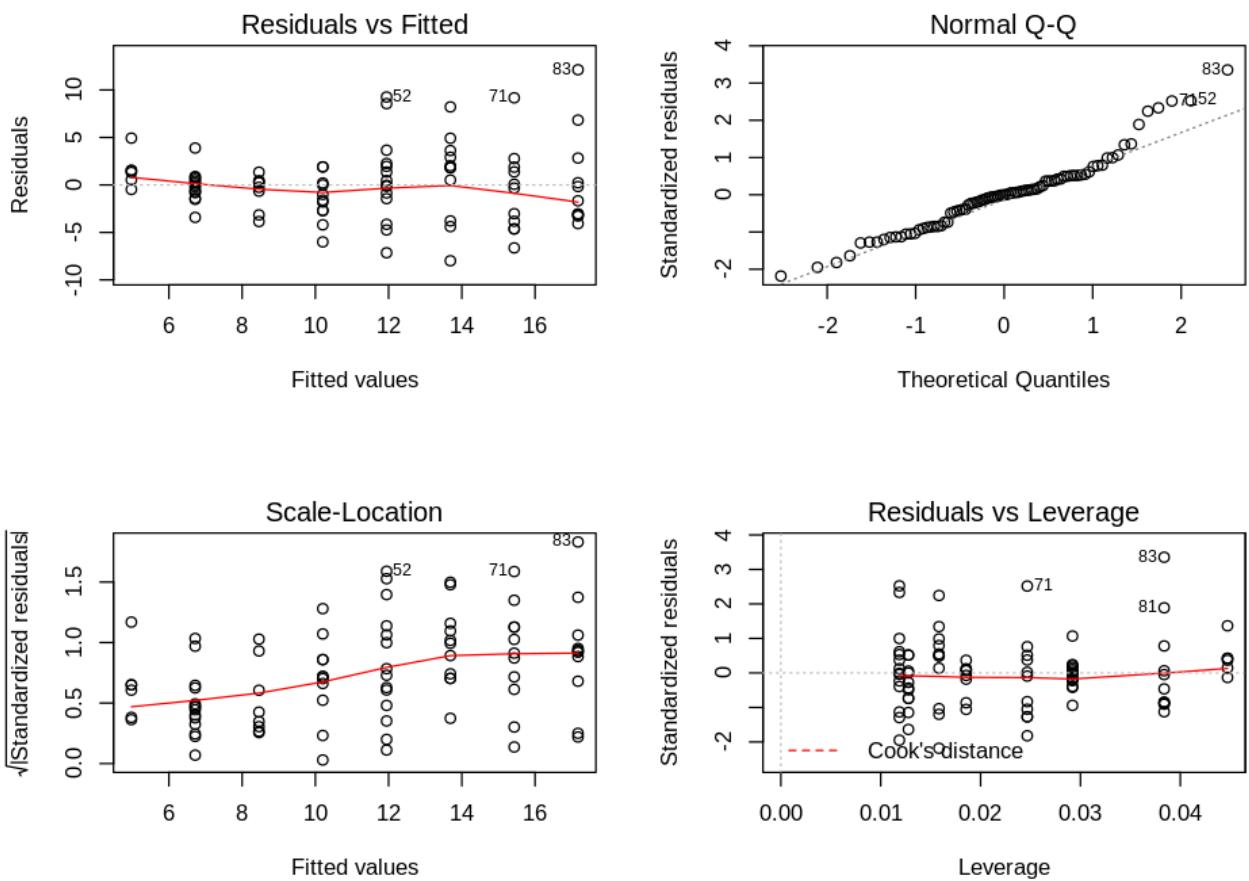
SLR Workshop Task: 4. d. Graph the regression equation and the data points.

```
In [13]: plot(ASD_State_4_SLR$Year, ASD_State_4_SLR$Prevalence)
abline(fit_model, col="blue", lwd=2)
```



SLR Workshop Task: 5. e. Identify potential influential observations (outliers).

```
In [14]: # library(repr)
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=6)
par(mfrow=c(2, 2))
plot(fit_model)
par(mfrow=c(1, 1))
```



[Tips] We notice:

- Based on **Residual vs Leverage** chart, there seems no potential influential observations (outliers)

SLR Workshop Task: 6. f. At the 5% significance level, do the data provide sufficient evidence to conclude that the slope of the population regression line is not 0 and, hence, that [Year] is useful as a predictor of ASD [Prevalence]?

```
In [15]: summary(fit_model)
```

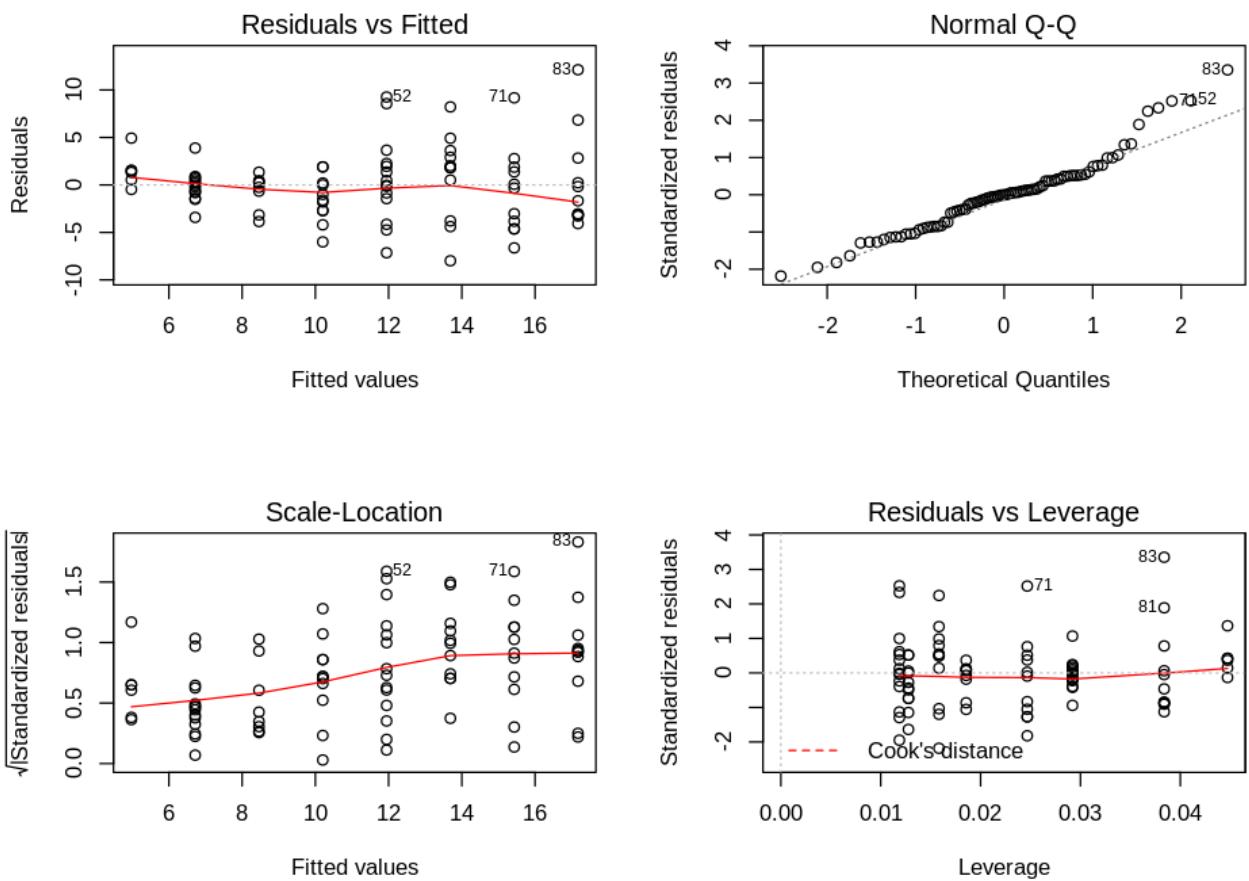
```
Call:  
lm(formula = Prevalence ~ Year, data = ASD_State_4_SLR)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-7.9888 -2.7032 -0.0104  1.7397 12.1255  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -1.738e+03  1.827e+02 -9.513 5.51e-15 ***  
Year          8.714e-01  9.101e-02   9.575 4.13e-15 ***  
---  
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
Residual standard error: 3.688 on 84 degrees of freedom  
Multiple R-squared:  0.5219,    Adjusted R-squared:  0.5162  
F-statistic: 91.69 on 1 and 84 DF,  p-value: 4.13e-15
```

[Tips] We notice:

2. F-test's p-value is 4.13e-15, which is smaller than 0.05, thus above 95% confidence.
-

SLR Workshop Task: 7. g. Obtain the residuals and create a residual plot. Decide whether it is reasonable to consider that the assumptions for regression analysis are met by the variables in questions.

```
In [16]: # library(repr)
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=6)
par(mfrow=c(2, 2))
plot(fit_model)
par(mfrow=c(1, 1))
```



[Tips] We notice:

- Based on **Residual vs Fitted**, **Sacle-Location**, and **Normal Q-Q** charts, the residuals (vs fitted) are following linear assumption, with slightly "fan-shape" at larger Year values (Heteroscedasticity).
<https://statisticsbyjim.com/regression/heteroscedasticity-regression/>
[\(https://statisticsbyjim.com/regression/heteroscedasticity-regression/\)](https://statisticsbyjim.com/regression/heteroscedasticity-regression/)
- We are to explore polynomial regression method for this issue later.

SLR Workshop Task: 8. h. Compute and interpret the coefficient of determination, R^2 .

```
In [17]: summary(fit_model)
```

```
Call:  
lm(formula = Prevalence ~ Year, data = ASD_State_4_SLR)  
  
Residuals:  
    Min      1Q  Median      3Q     Max  
-7.9888 -2.7032 -0.0104  1.7397 12.1255  
  
Coefficients:  
            Estimate Std. Error t value Pr(>|t|)  
(Intercept) -1.738e+03  1.827e+02 -9.513 5.51e-15 ***  
Year          8.714e-01  9.101e-02   9.575 4.13e-15 ***  
---  
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1  
  
Residual standard error: 3.688 on 84 degrees of freedom  
Multiple R-squared:  0.5219,    Adjusted R-squared:  0.5162  
F-statistic: 91.69 on 1 and 84 DF,  p-value: 4.13e-15
```

[Tips] We notice:

- R^2 is 0.5219
- Adjusted R^2 is 0.5162

SLR Workshop Task: 9. i. Find the predicted ASD Prevalence of future Year.

```
In [18]: future_year = 2025  
newdata = data.frame(Year = future_year)  
predict(fit_model, newdata)  
#  
cat("Predicted ASD Prevalence of Year [", future_year, "] is", round(predict(fit
```

```
1: 26.7599815890101
```

Predicted ASD Prevalence of Year [2025] is 26.8 per 1,000 Children

SLR Workshop Task: 10. j. Determine a 95% confidence interval for the predicted ASD Prevalence.

```
In [19]: predict(fit_model, newdata, interval = "predict")
```

fit	lwr	upr
26.75998	18.72351	34.79645

```
In [20]: cat("\nPredicted ASD Prevalence of Year [", future_year, "] (95% Upper CI) is  
round(predict(fit_model,newdata, interval = "predict")[3], 1), \"per 1,000  
  
cat("\nPredicted ASD Prevalence of Year [", future_year, "] (95% Lower CI) is  
round(predict(fit_model,newdata, interval = "predict")[2], 1), \"per 1,000
```

Predicted ASD Prevalence of Year [2025] (95% Upper CI) is 34.8 per 1,000 Children
Predicted ASD Prevalence of Year [2025] (95% Lower CI) is 18.7 per 1,000 Children

Quiz:

Create Prevalence ~ Year SLR model for Data Source: SPED

```
In [21]: # Write your code below and press Shift+Enter to execute
```

Double-click **here** for the solution.

.0

Linear Model: Multiple Linear Regression (MLR)

Linear Model: Multiple Linear Regression (MLR) - Workshop Task

Workshop Task:

1. a. Get the data.
2. b. Discover and visualize the data to gain insights (Is there missing Value in the dataframe, then how to deal with the missing value)
3. c. Visualize Data and trends
4. d. Compute correlation between variables and apply multiple regression.
5. e. Check multicollinearity, then how to remove multicollinearity.
6. f. How is your final model looks like?

MLR Workshop Task: 1. a. Get the data.

Use Case Data: "../dataset/ADV_ASD_State_R.csv"

Read in CSV data, storing as R dataframe

```
In [22]: # Read back in above saved file:  
# ASD_State <- read.csv("../dataset/ADV_ASD_State_R.csv")  
# ASD_State$Year_Factor <- factor(ASD_State$Year_Factor, ordered = TRUE) # Con  
# ASD_State$Prevalence_Risk2 = factor(ASD_State$Prevalence_Risk2, ordered=TRUE)  
# ASD_State$Prevalence_Risk4 = factor(ASD_State$Prevalence_Risk4, ordered=TRUE)
```

```
In [23]: head(ASD_State)
```

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_F
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Ariz
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Geo
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	Marylan

```
In [24]: names(ASD_State)
```

```
'State' 'Denominator' 'Prevalence' 'Lower.CI' 'Upper.CI' 'Year' 'Source' 'Source_Full1'  
'State_Full1' 'State_Full2' 'Numerator_ASD' 'Numerator_NonASD' 'Proportion'  
'Chi_Wilson_Corrected_Lower.CI' 'Chi_Wilson_Corrected_Upper.CI' 'Male.Prevalence'  
'Male.Lower.CI' 'Male.Upper.CI' 'Female.Prevalence' 'Female.Lower.CI' 'Female.Upper.CI'  
'Non.hispanic.white.Prevalence' 'Non.hispanic.white.Lower.CI' 'Non.hispanic.white.Upper.CI'  
'Non.hispanic.black.Prevalence' 'Non.hispanic.black.Lower.CI' 'Non.hispanic.black.Upper.CI'  
'Hispanic.Prevalence' 'Hispanic.Lower.CI' 'Hispanic.Upper.CI' 'Asian.or.Pacific.Islander.Prevalence'  
'Asian.or.Pacific.Islander.Lower.CI' 'Asian.or.Pacific.Islander.Upper.CI' 'State_Region' 'Source_UC'  
'Source_Full3' 'Prevalence_Risk2' 'Prevalence_Risk4' 'Year_Factor'
```

```
In [25]: # Filter to include relevant columns for MLR:
# Dependent variable: Prevalence
# independent variable: Let's include all at the moment
ASD_State_4_MLR = ASD_State
#
dim(ASD_State_4_MLR)
head(ASD_State_4_MLR)
```

1692 39

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_F
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizo
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Geo
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	Marylan

MLR Workshop Task: 2. b. Discover and visualize the data to gain insights (Is there missing Value in the dataframe, then how to deal with the missing value).

```
In [26]: summary(ASD_State_4_MLR)
```

State	Denominator	Prevalence	Lower.CI	
AZ	Min. : 40	Min. : 0.400	Min. : 0.30	
MD	1st Qu.: 40	1st Qu.: 3.100	1st Qu.: 2.80	
GA	Median : 39	Median : 5.600	Median : 5.30	
MO	Mean : 39	Mean : 7.191	Mean : 6.42	
NC	3rd Qu.: 39	3rd Qu.: 9.200	3rd Qu.: 8.60	
WI	Max. : 39	Max. : 42.700	Max. : 29.90	
(Other):1456				
		Upper.CI	Year	Source
		Min. : 0.600	Min. : 2000	addm: 86
		1st Qu.: 3.300	1st Qu.: 2003	medi:655
		Median : 5.900	Median : 2007	nsch: 98
		Mean : 8.262	Mean : 2007	sped:853
		3rd Qu.: 9.700	3rd Qu.: 2011	
		Max. : 69.000	Max. : 2016	
				Source_Full1
				Autism & Developmental Disabilities Monitoring Network: 86
				Medicaid :655
				National Survey of Children's Health : 98

```
In [27]: # Check whether each columns got missing value:  
lapply(ASD_State_4_MLR, function(col_x)sum(is.na(col_x)))  
  
# Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=3)  
barplot(apply(ASD_State_4_MLR, 2, function(col_x)sum(is.na(col_x))))
```

```
$State  
0  
$Denominator  
0  
$Prevalence  
0  
$Lower.CI  
0  
$Upper.CI  
0  
$Year  
0  
$Source  
0  
$Source_Full1  
0  
$State_Full1
```

```
In [28]: dim(ASD_State_4_MLR)
```

```
1692 39
```

```
In [29]: #Get all the column variables which contains missing value  
NA_Column_Names <- names(ASD_State_4_MLR[0, colSums(is.na(ASD_State_4_MLR)) >  
#  
NA_Column_Names
```

```
'Male.Prevalence' 'Male.Lower.CI' 'Male.Upper.CI' 'Female.Prevalence' 'Female.Lower.CI'  
'Female.Upper.CI' 'Non.hispanic.white.Prevalence' 'Non.hispanic.white.Lower.CI'  
'Non.hispanic.white.Upper.CI' 'Non.hispanic.black.Prevalence' 'Non.hispanic.black.Lower.CI'  
'Non.hispanic.black.Upper.CI' 'Hispanic.Prevalence' 'Hispanic.Lower.CI' 'Hispanic.Upper.CI'  
'Asian.or.Pacific.Islander.Prevalence' 'Asian.or.Pacific.Islander.Lower.CI'  
'Asian.or.Pacific.Islander.Upper.CI'
```

```
In [30]: # Remove these columns from dataframe
ASD_State_4_MLR <- ASD_State_4_MLR[ , !(names(ASD_State_4_MLR) %in% NA_Column_#]
# head(ASD_State_4_MLR)
```

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_F
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizo
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Geo
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	Marylan

No missing values, as they have been handled earlier. **Hurrah!**

But some varialbe contains "leaky" information, which can be used to directly calculate the dependent variable: Prevalence. This won't happen in real world scenario, thus they need to be removed.

```
In [31]: cbind(names(ASD_State_4_MLR), c(1:length(names(ASD_State_4_MLR))))
```

State	1
Denominator	2
Prevalence	3
Lower.CI	4
Upper.CI	5
Year	6
Source	7
Source_Full1	8
State_Full1	9
State_Full2	10
Numerator_ASD	11
Numerator_NonASD	12
Proportion	13

```
In [32]: Leaky_Column_Names = c('Lower.CI', 'Upper.CI', 'Numerator_ASD', 'Numerator_NonASD', 'Chi_Wilson_Corrected_Lower.CI', 'Chi_Wilson_Corrected_Upper.CI', 'Prevalence_Risk2', 'Prevalence_Risk4')
```

```
In [33]: # Remove these columns from dataframe
ASD_State_4_MLR <- ASD_State_4_MLR[ , !(names(ASD_State_4_MLR) %in% Leaky_Colu
#
head(ASD_State_4_MLR)
```

State	Denominator	Prevalence	Year	Source	Source_Full1	State_Full1	State_Full2	State_Region
AZ	45322	6.5	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizona	D8 Mountain
GA	43593	6.5	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Georgia	D5 South Atlantic
MD	21532	5.5	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	MD-Maryland	D5 South Atlantic

Remove redundant/duplicate variables (aliased coefficients), retaining one for each type of information is enough:

<https://en.wikipedia.org/wiki/Multicollinearity> (<https://en.wikipedia.org/wiki/Multicollinearity>)

<https://stats.stackexchange.com/questions/112442/what-are-aliased-coefficients>
[\(<https://stats.stackexchange.com/questions/112442/what-are-aliased-coefficients>\)](https://stats.stackexchange.com/questions/112442/what-are-aliased-coefficients)

```
In [34]: Redundant_Column_Names = c('State', 'Source_Full1', 'State_Full1', 'State_Regis
```

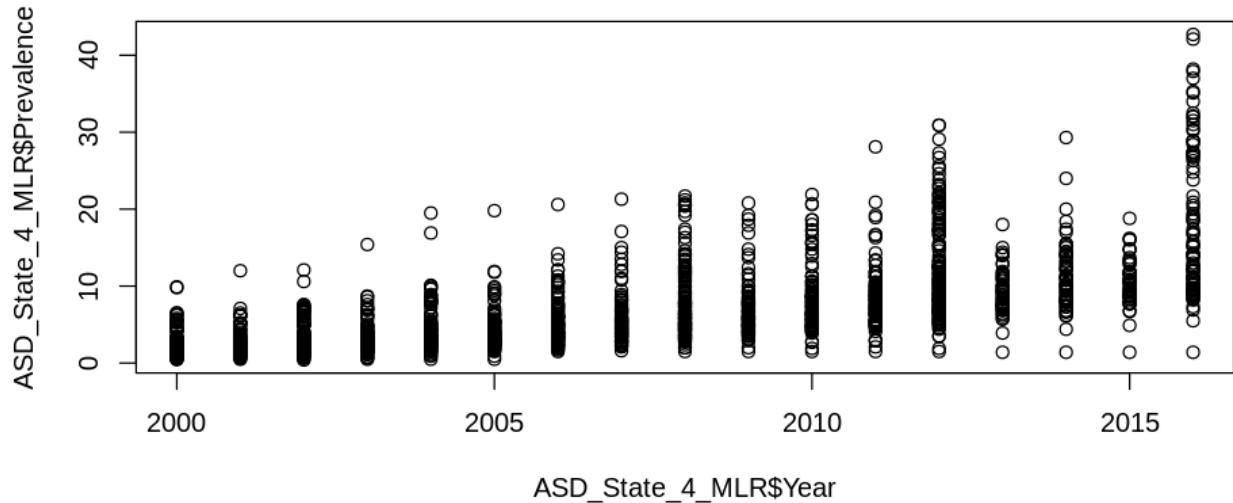
```
In [35]: # Remove these columns from dataframe
ASD_State_4_MLR <- ASD_State_4_MLR[ , !(names(ASD_State_4_MLR) %in% Redundant_
#
head(ASD_State_4_MLR)
```

Denominator	Prevalence	Year	Source	State_Full2
45322	6.5	2000	addm	AZ-Arizona
43593	6.5	2000	addm	GA-Georgia
21532	5.5	2000	addm	MD-Maryland
29714	9.9	2000	addm	NJ-New Jersey
24535	6.3	2000	addm	SC-South Carolina
23065	4.5	2000	addm	WV-West Virginia

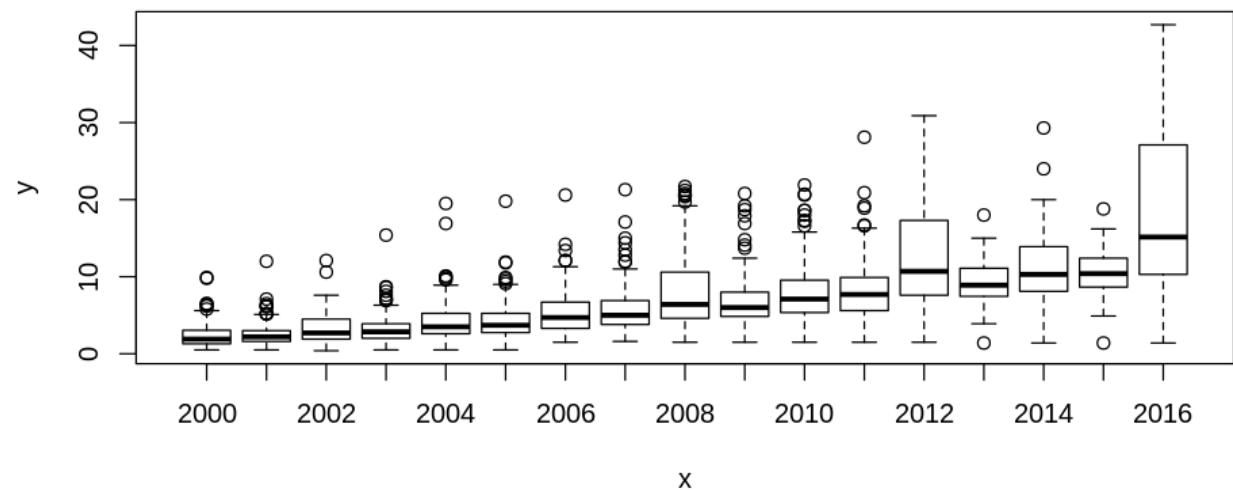
MLR Workshop Task: 3. c. Visualize the data to gain insights

```
In [36]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)
```

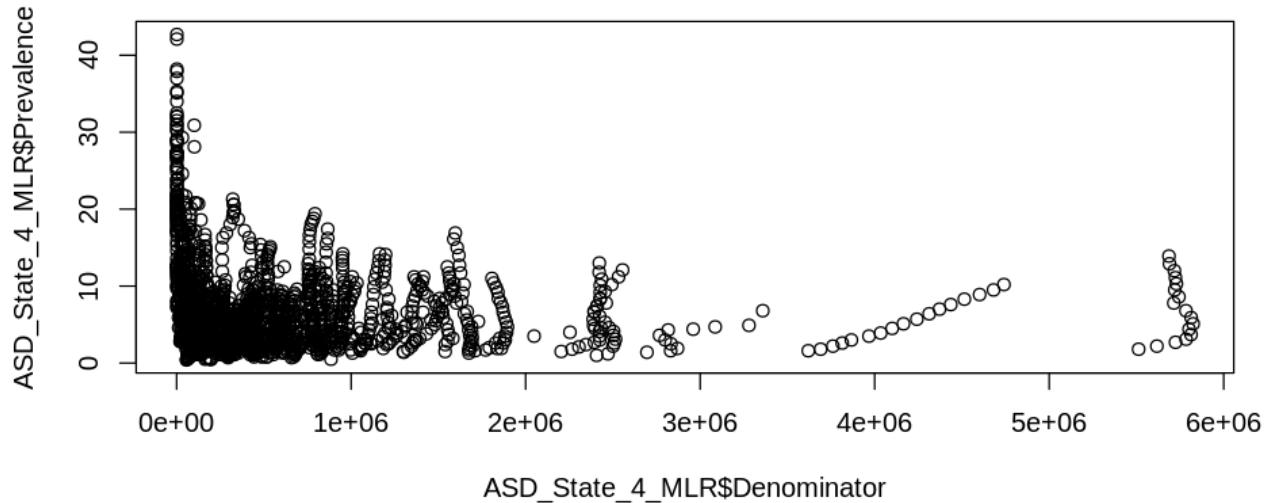
```
In [37]: plot(ASD_State_4_MLR$Year, ASD_State_4_MLR$Prevalence)
```



```
In [38]: plot(as.factor(ASD_State_4_MLR$Year), ASD_State_4_MLR$Prevalence)
```

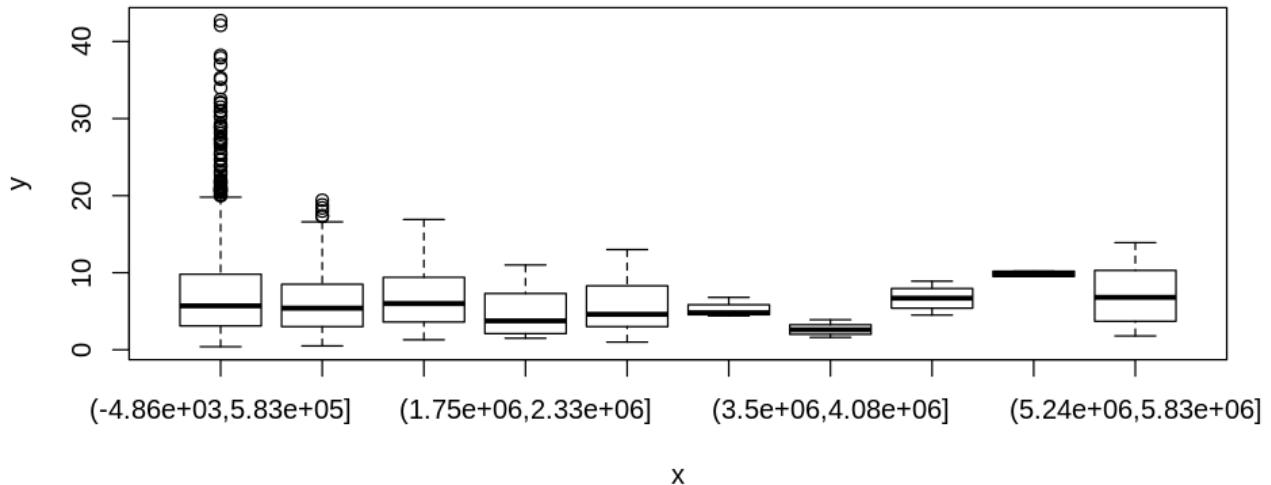


```
In [39]: plot(ASD_State_4_MLR$Denominator, ASD_State_4_MLR$Prevalence)
```

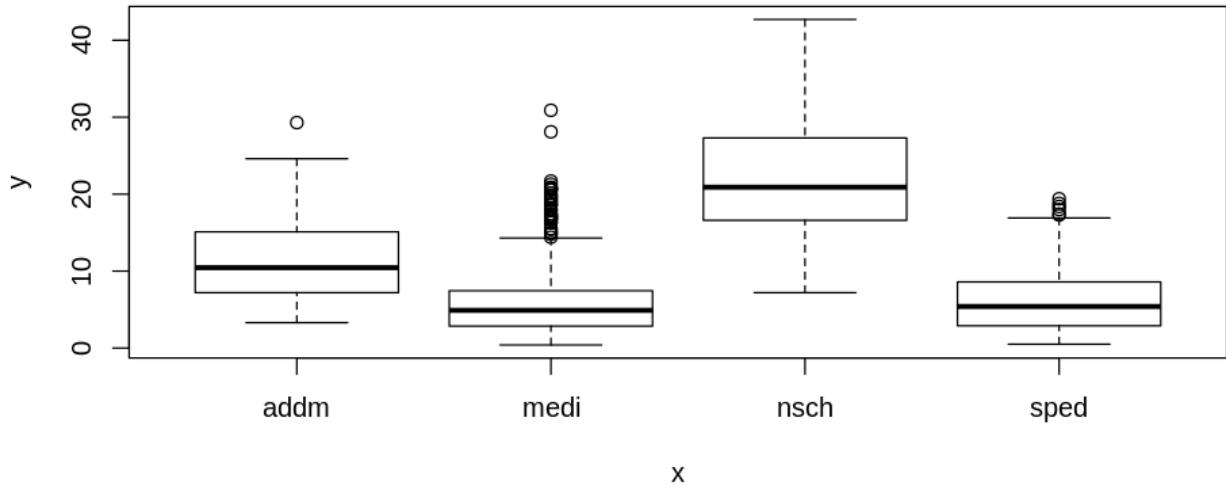


```
In [40]: # To use bin() function  
# https://www.rdocumentation.org/packages/OneR/versions/2.2/topics/bin  
if(!require(OneR)){install.packages("OneR")}  
library('OneR')  
  
# Bin 'Denominator'  
plot(bin(ASD_State_4_MLR$Denominator, nbins = 10), ASD_State_4_MLR$Prevalence)
```

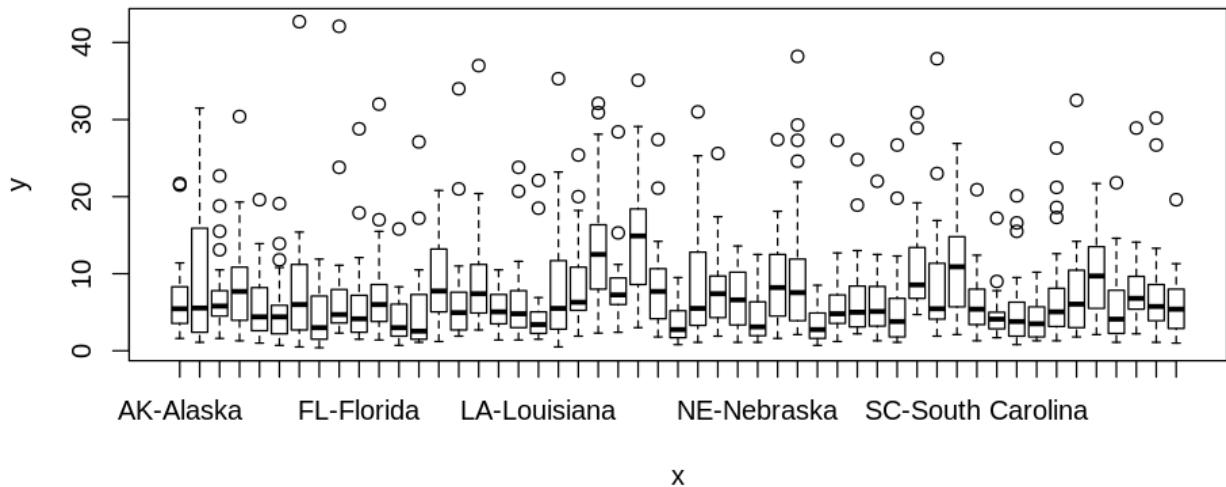
Loading required package: OneR



```
In [41]: plot(ASD_State_4_MLR$Source, ASD_State_4_MLR$Prevalence)
```



```
In [42]: plot(ASD_State_4_MLR$State_Full2, ASD_State_4_MLR$Prevalence)
```



MLR Workshop Task: 4. d. Compute correlation between variables and apply multiple regression.

Recode categorical variable to dummy (numeric) variable using one-hot encoding:

```
In [43]: # To use select_if() function
if(!require(dplyr)){install.packages("dplyr")}
library("dplyr")

summary(select_if(ASD_State_4_MLR, is.numeric))
```

Loading required package: dplyr

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

Denominator	Prevalence	Year
Min. : 965	Min. : 0.400	Min. :2000
1st Qu.: 107151	1st Qu.: 3.100	1st Qu.:2003
Median : 353328	Median : 5.600	Median :2007
Mean : 604689	Mean : 7.191	Mean :2007
3rd Qu.: 767928	3rd Qu.: 9.200	3rd Qu.:2011
Max. :5824922	Max. :42.700	Max. :2016

```
In [44]: correlation = cor(select_if(ASD_State_4_MLR, is.numeric))
correlation
```

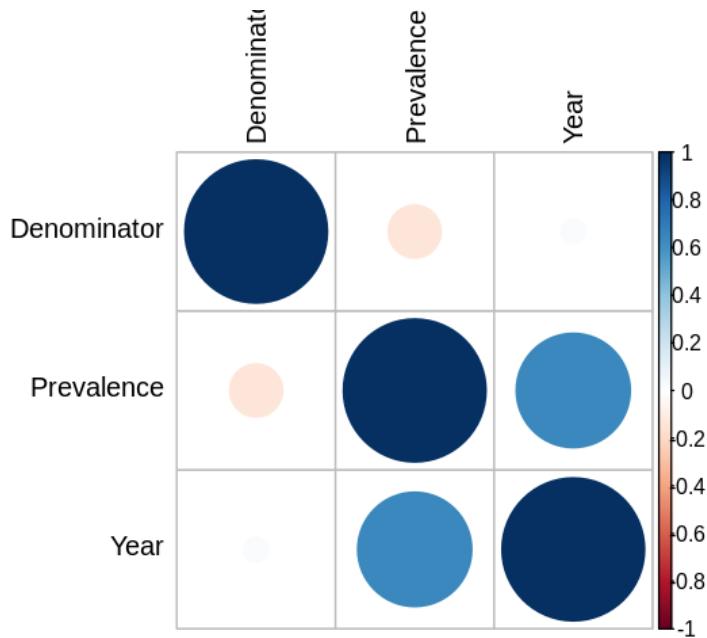
	Denominator	Prevalence	Year
Denominator	1.00000000	-0.1374662	0.02851671
Prevalence	-0.13746621	1.0000000	0.64002950
Year	0.02851671	0.6400295	1.00000000

```
In [45]: # Variable's correlation against target dependent variable:
correlation[, 2]
```

Denominator	-0.137466206927898
Prevalence	1
Year	0.640029495747179

```
In [46]: if(!require(corrplot)){install.packages("corrplot")}  
library("corrplot")  
corrplot(correlation, tl.col="black", tl.pos = "lt")
```

Loading required package: corrplot
corrplot 0.84 loaded



```
In [47]: str(ASD_State_4_MLR)
```

```
'data.frame': 1692 obs. of 5 variables:  
 $ Denominator: int 45322 43593 21532 29714 24535 23065 35472 45113 36472 11  
020 ...  
 $ Prevalence : num 6.5 6.5 5.5 9.9 6.3 4.5 3.3 6.2 6.9 5.9 ...  
 $ Year       : int 2000 2000 2000 2000 2000 2000 2002 2002 2002 2002 ...  
 $ Source     : Factor w/ 4 levels "addm","medi",...: 1 1 1 1 1 1 1 1 1 1 ...  
 $ State_Full2: Factor w/ 51 levels "AK-Alaska","AL-Alabama",...: 4 11 21 32 4  
1 50 2 4 3 6 ...
```

```
In [48]: # To build (National level) ASD Prevalence predictive model for all state's:  
# In situations that we won't know the US. State name, we can also fit a model  
fit_model = lm(Prevalence ~ . - State_Full2, data = ASD_State_4_MLR) # Exclude  
#  
summary(fit_model)
```

Call:
`lm(formula = Prevalence ~ . - State_Full2, data = ASD_State_4_MLR)`

Residuals:

	Min	1Q	Median	3Q	Max
	-11.4476	-1.9332	-0.2786	1.2479	21.0130

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.310e+03	3.819e+01	-34.306	<2e-16 ***
Denominator	-1.139e-07	1.057e-07	-1.078	0.281
Year	6.583e-01	1.902e-02	34.606	<2e-16 ***
Sourcemedi	-4.550e+00	3.964e-01	-11.478	<2e-16 ***
Sourcensch	6.699e+00	5.171e-01	12.956	<2e-16 ***
Sourcesped	-5.611e+00	3.984e-01	-14.085	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.43 on 1686 degrees of freedom
Multiple R-squared: 0.6585, Adjusted R-squared: 0.6575
F-statistic: 650.2 on 5 and 1686 DF, p-value: < 2.2e-16

Adjusted $R^2 = 0.6575$

```
In [49]: # To build (US. State level) ASD Prevalence predictive model for specific stat  
# In situations that we shall know the US. State name. (A state name is requir  
fit_model = lm(Prevalence ~ . , data = ASD_State_4_MLR) # "~." means all other  
#  
summary(fit_model)
```

Call:
`lm(formula = Prevalence ~ . , data = ASD_State_4_MLR)`

Residuals:

	Min	1Q	Median	3Q	Max
	-11.3326	-1.3626	-0.0689	1.2558	19.0273

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.305e+03	3.144e+01	-41.498	< 2e-16 *
**				
Denominator	1.152e-06	2.252e-07	5.115	3.50e-07 *
**				
Year	6.558e-01	1.566e-02	41.889	< 2e-16 *
**				
Sourcemedi	-4.997e+00	3.557e-01	-14.048	< 2e-16 *
**				
Sourcensch	6.100e+00	4.404e-01	13.853	< 2e-16 *

Adjusted $R^2 = 0.7697$

MLR Workshop Task: 5. e. Check multicollinearity, then how to remove multicollinearity.

< Detection of multicollinearity >

Some authors have suggested a formal detection-tolerance or the variance inflation factor (VIF) for multicollinearity. A VIF of 5 or 10 and above indicates a multicollinearity problem.

```
In [50]: # To use select_if() function  
if(!require(car)){install.packages("car")}  
library("car")
```

Loading required package: car
Loading required package: carData

Attaching package: 'car'

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

recode

```
In [51]: vif(fit_model)
```

	GVIF	Df	GVIF^(1/(2*Df))
Denominator	7.737921	1	2.781712
Year	1.146987	1	1.070975
Source	2.502245	3	1.165167
State_Full2	7.768262	50	1.020712

[Tips] We notice VIF of Denominator and State_Full2 are high. Let's exclude them one at a time.

Retain: State; Remove: Denominator then re-build model:

```
In [52]: # To build (National level) ASD Prevalence predictive model for all state's:  
# In situations that we won't know the US. State name, we can also fit a model  
fit_model_with_State = lm(Prevalence ~ . - Denominator, data = ASD_State_4_MLR)  
#  
summary(fit_model_with_State)
```

Call:
`lm(formula = Prevalence ~ . - Denominator, data = ASD_State_4_MLR)`

Residuals:

	Min	10	Median	30	Max
	-10.7769	-1.3853	-0.0184	1.2957	19.2545

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.319e+03	3.157e+01	-41.774	< 2e-16 *
Year	6.624e-01	1.572e-02	42.130	< 2e-16 *
Sourcemedi	-4.503e+00	3.450e-01	-13.053	< 2e-16 *
Sourcensch	6.149e+00	4.436e-01	13.861	< 2e-16 *
-----	-----	-----	-----	-----

```
In [53]: vif(fit_model_with_State)
```

	GVIF	Df	GVIF^(1/(2*Df))
Year	1.139247	1	1.067355
Source	1.306350	3	1.045546
State_Full2	1.149997	50	1.001399

Adjusted $R^2 = 0.7662$

Retain: Denominator; Remove: State; then re-build model:

```
In [54]: # To build (National level) ASD Prevalence predictive model for all state's:  
# In situations that we won't know the US. State name, we can also fit a model  
fit_model_with_Denominator = lm(Prevalence ~ . - State_Full2, data = ASD_State  
#  
summary(fit_model_with_Denominator)
```

Call:
`lm(formula = Prevalence ~ . - State_Full2, data = ASD_State_4_MLR)`

Residuals:

Min	10	Median	30	Max
-11.4476	-1.9332	-0.2786	1.2479	21.0130

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1.310e+03	3.819e+01	-34.306	<2e-16 ***
Denominator	-1.139e-07	1.057e-07	-1.078	0.281
Year	6.583e-01	1.902e-02	34.606	<2e-16 ***
Sourcemedi	-4.550e+00	3.964e-01	-11.478	<2e-16 ***
Sourcensch	6.699e+00	5.171e-01	12.956	<2e-16 ***
Sourcesped	-5.611e+00	3.984e-01	-14.085	<2e-16 ***

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

Residual standard error: 3.43 on 1686 degrees of freedom
Multiple R-squared: 0.6585, Adjusted R-squared: 0.6575
F-statistic: 650.2 on 5 and 1686 DF, p-value: < 2.2e-16

```
In [55]: vif(fit_model_with_Denominator)
```

	GVIF	Df	GVIF^(1/(2*Df))
Denominator	1.145505	1	1.070283
Year	1.138674	1	1.067086
Source	1.301973	3	1.044962

Adjusted $R^2 = 0.6575$

MLR Workshop Task: 6. f. How is your final model looks like?

- During prediction, if US. State name will be known, then the `fit_model_with_State` can be better because it has higher R^2 value.
- During prediction, if US. State name will NOT be known, then the `fit_model_with_Denominator` can be adopted because it doesn't require state name as input for prediction.

MLR Prediciton 1

Let's use `fit_model_with_State` to predict CA-California ASD Prevalence of Year 2016 if ADDM would have conducted a survey

```
In [56]: newdata = ASD_State_4_MLR[1,] # Copy datastructure
newdata$Prevalence = NA
newdata$Denominator = 50000
newdata$Year = 2016
newdata$Source = "addm"
#newdata$State_Full2 = "CA-California"
newdata$State_Full2 = "AZ-Arizona"

newdata
```

Denominator	Prevalence	Year	Source	State_Full2
50000	NA	2016	addm	AZ-Arizona

```
In [57]: predict(fit_model_with_State, newdata, interval = "predict")
#
cat("Predicted ASD Prevalence is", round(predict(fit_model_with_State, newdata
```

fit	lwr	upr
17.54292	11.88535	23.20049

Predicted ASD Prevalence is 17.5 per 1,000 Children

MLR Prediciton 2

Let's use **fit_model_with_Denominator** to predict National level ASD Prevalence of Year 2016 if ADDM would have conducted a survey

```
In [58]: predict(fit_model_with_Denominator, newdata, interval = "predict")
#
cat("Predicted ASD Prevalence is", round(predict(fit_model_with_Denominator, n
```

fit	lwr	upr
17.07629	10.30306	23.84952

Predicted ASD Prevalence is 17.1 per 1,000 Children

MLR Prediciton 2

Let's use **fit_model** to predict FL-Florida State level ASD Prevalence of Year 2025 if SPED will conduct a record review/survey of 2,600,000 children.

```
In [59]: newdata = ASD_State_4_MLR[1,] # Copy datastructure
newdata$Prevalence = NA
newdata$Denominator = 2600000
newdata$Year = 2025
newdata$Source = "sped"
newdata$State_Full2 = "FL-Florida"

newdata
```

Denominator	Prevalence	Year	Source	State_Full2
2600000	NA	2025	sped	FL-Florida

```
In [60]: predict(fit_model, newdata, interval = "predict")
#
cat("Predicted ASD Prevalence is", round(predict(fit_model, newdata), 1), "per
```

fit	lwr	upr
16.63773	11.00985	22.2656

Predicted ASD Prevalence is 16.6 per 1,000 Children

.0

Linear Model: Polynomial (Linear) Regression (PLR)

Linear Model: Polynomial (Linear) Regression (PLR) - Workshop Task

Workshop Task:

1. a. Get the data.
2. b. Discover and visualize the data to gain insights (Is there missing Value in the dataframe, then how to deal with the missing value)
3. c. Visualize Data and trends
4. d. Compute correlation between variables and apply multiple regression.
5. e. Multiple polynomial regression.

PLR Workshop Task: 1. a. Get the data.

Use Case Data: ["../dataset/ADV_ASD_State_R.csv"](#)

Read in CSV data, storing as R dataframe

```
In [61]: # Read back in above saved file:
ASD_State <- read.csv("../dataset/ADV_ASD_State_R.csv")
ASD_State$Year_Factor <- factor(ASD_State$Year_Factor, ordered = TRUE) # Conve
ASD_State$Prevalence_Risk2 = factor(ASD_State$Prevalence_Risk2, ordered=TRUE,
ASD_State$Prevalence_Risk4 = factor(ASD_State$Prevalence_Risk4, ordered=TRUE,
```

In [62]: head(ASD_State)

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_Full2
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Ariz
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Geo
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	Marylan

In [63]: names(ASD_State)

```
'State' 'Denominator' 'Prevalence' 'Lower.CI' 'Upper.CI' 'Year' 'Source' 'Source_Full1'  
'State_Full1' 'State_Full2' 'Numerator_ASD' 'Numerator_NonASD' 'Proportion'  
'Chi_Wilson_Corrected_Lower.CI' 'Chi_Wilson_Corrected_Upper.CI' 'Male.Prevalence'  
'Male.Lower.CI' 'Male.Upper.CI' 'Female.Prevalence' 'Female.Lower.CI' 'Female.Upper.CI'  
'Non.hispanic.white.Prevalence' 'Non.hispanic.white.Lower.CI' 'Non.hispanic.white.Upper.CI'  
'Non.hispanic.black.Prevalence' 'Non.hispanic.black.Lower.CI' 'Non.hispanic.black.Upper.CI'  
'Hispanic.Prevalence' 'Hispanic.Lower.CI' 'Hispanic.Upper.CI' 'Asian.or.Pacific.Islander.Prevalence'  
'Asian.or.Pacific.Islander.Lower.CI' 'Asian.or.Pacific.Islander.Upper.CI' 'State_Region' 'Source_UC'  
'Source_Full3' 'Prevalence_Risk2' 'Prevalence_Risk4' 'Year_Factor'
```

In [64]: # Filter [Source: SPED], including only two columns for SLR:

```
# Dependent variable: Prevalence  
# independent variable: Year  
# ASD_State_4_PLR = subset(ASD_State, Source_UC == 'SPED' & State_Full2 == 'MN')  
# ASD_State_4_PLR = subset(ASD_State, Source_UC == 'SPED' & State_Full2 == 'MS')  
ASD_State_4_PLR = subset(ASD_State, Source_UC == 'SPED' & State_Full2 == 'FL-F')  
#  
dim(ASD_State_4_PLR)  
head(ASD_State_4_PLR)
```

17 2

	Prevalence	Year
849	1.5	2000
900	1.8	2001
951	2.1	2002
1002	2.4	2003
1052	2.7	2004
1100	3.0	2005

PLR Workshop Task: 2. b. Discover and visualize the data to gain insights (Is there missing Value in the dataframe, then how to deal with the missing value).

In [65]: `summary(ASD_State_4_PLR)`

Prevalence	Year
Min. : 1.500	Min. :2000
1st Qu.: 2.700	1st Qu.:2004
Median : 4.900	Median :2008
Mean : 5.694	Mean :2008
3rd Qu.: 8.300	3rd Qu.:2012
Max. :12.100	Max. :2016

In [66]: `# Check whether each columns got missing value:
lapply(ASD_State_4_PLR, function(col_x)sum(is.na(col_x)))

Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=3)
barplot(apply(ASD_State_4_PLR, 2, function(col_x)sum(is.na(col_x))))`

\$Prevalence

0

\$Year

0

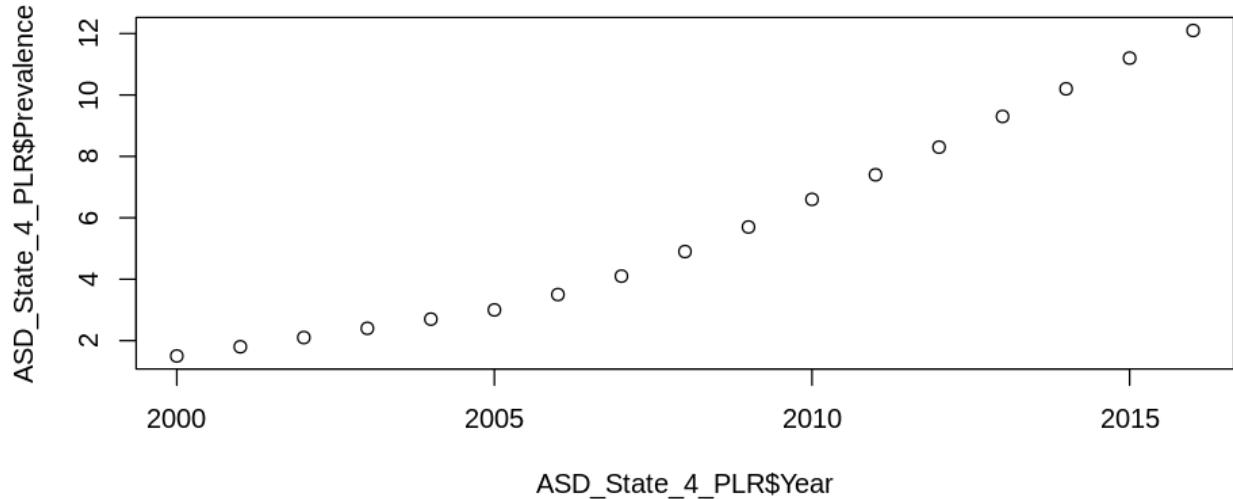


No missing values

PLR Workshop Task: 3. c. Visualize the data to gain insights

In [67]: `# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)`

```
In [68]: plot(ASD_State_4_PLR$Year, ASD_State_4_PLR$Prevalence)
```



PLR Workshop Task: 4. d. Compute correlation between variables and apply multiple regression.

Recode categorical variable to dummy (numeric) variable using one-hot encoding:

```
In [69]: # To use select_if() function
if(!require(dplyr)){install.packages("dplyr")}
library("dplyr")

summary(select_if(ASD_State_4_PLR, is.numeric))
```

Prevalence	Year
Min. : 1.500	Min. :2000
1st Qu.: 2.700	1st Qu.:2004
Median : 4.900	Median :2008
Mean : 5.694	Mean :2008
3rd Qu.: 8.300	3rd Qu.:2012
Max. :12.100	Max. :2016

```
In [70]: correlation = cor(select_if(ASD_State_4_PLR, is.numeric))
correlation
```

Prevalence	Year
Prevalence	1.000000 0.980119
Year	0.980119 1.000000

```
In [71]: # Variable's correlation against target dependent variable:
correlation[, 1]
```

Prevalence	1
Year	0.980118967682229

```
In [72]: str(ASD_State_4_PLR)
```

```
'data.frame': 17 obs. of 2 variables:  
 $ Prevalence: num 1.5 1.8 2.1 2.4 2.7 3 3.5 4.1 4.9 5.7 ...  
 $ Year       : int 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 ...
```

```
In [73]: # SLR  
fit_model_SLR = lm(Prevalence ~ Year, data = ASD_State_4_PLR)  
#  
summary(fit_model_SLR)
```

Call:

```
lm(formula = Prevalence ~ Year, data = ASD_State_4_PLR)
```

Residuals:

Min	1Q	Median	3Q	Max
-0.9150	-0.6566	-0.1108	0.4809	1.2392

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-1358.0726	71.2824	-19.05	6.37e-12 ***
Year	0.6792	0.0355	19.13	6.00e-12 ***

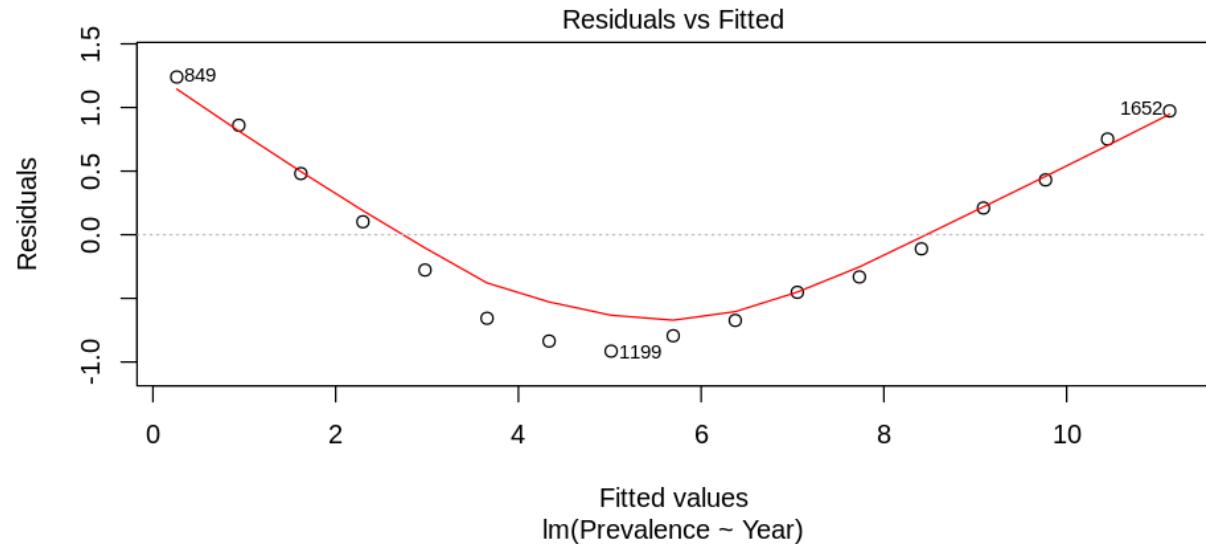
Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'
	0.1 ' '	1		

Residual standard error: 0.717 on 15 degrees of freedom

Multiple R-squared: 0.9606, Adjusted R-squared: 0.958

F-statistic: 366 on 1 and 15 DF, p-value: 5.995e-12

```
In [74]: plot(fit_model_SLR)
```



Adjusted $R^2 = 0.958$

```
In [75]: # PLR (quadratic)
fit_model_PLR = lm(Prevalence ~ Year + I(Year^2), data = ASD_State_4_PLR)
#
summary(fit_model_PLR)
```

Call:
`lm(formula = Prevalence ~ Year + I(Year^2), data = ASD_State_4_PLR)`

Residuals:

Min	1Q	Median	3Q	Max
-0.26223	-0.05325	0.03671	0.11045	0.17918

Coefficients:

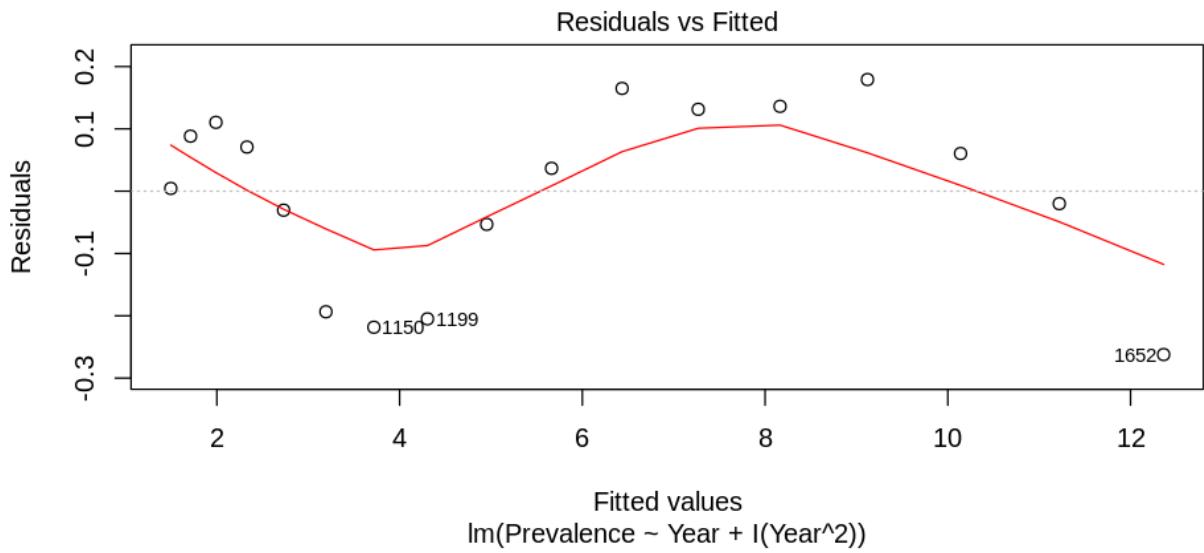
	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	1.231e+05	6.980e+03	17.64	5.87e-11 ***
Year	-1.233e+02	6.953e+00	-17.73	5.45e-11 ***
I(Year^2)	3.087e-02	1.731e-03	17.83	5.07e-11 ***

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1524 on 14 degrees of freedom
Multiple R-squared: 0.9983, Adjusted R-squared: 0.9981
F-statistic: 4209 on 2 and 14 DF, p-value: < 2.2e-16

About `I()` function: <https://stackoverflow.com/questions/8055508/in-r-formulas-why-do-i-have-to-use-the-i-function-on-power-terms-like-y-i> (<https://stackoverflow.com/questions/8055508/in-r-formulas-why-do-i-have-to-use-the-i-function-on-power-terms-like-y-i>).

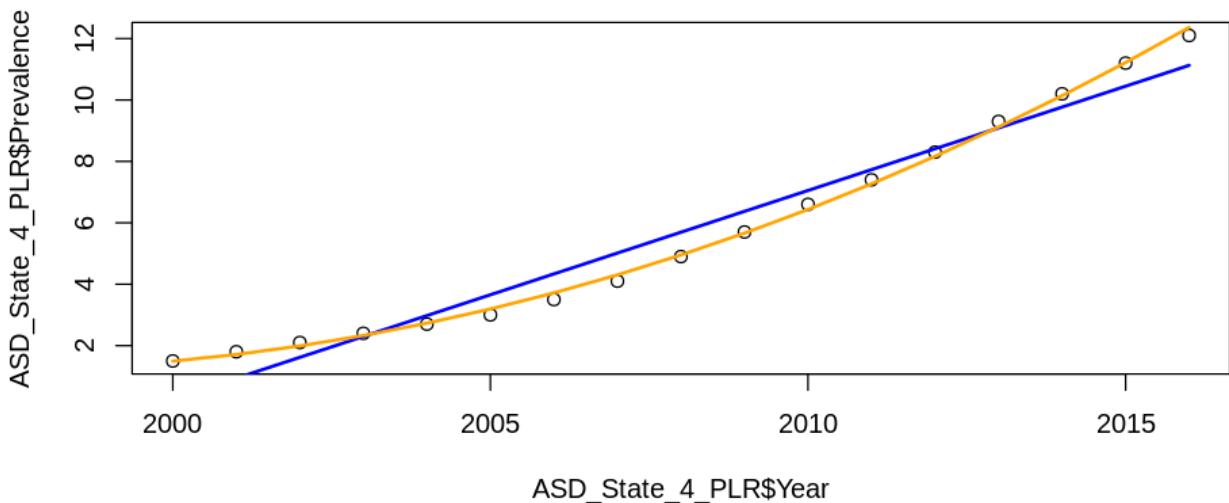
```
In [76]: plot(fit_model_PLR)
```



Adjusted $R^2 = 0.9981$

Visualise the difference between SLR & PLR:

```
In [77]: plot(ASD_State_4_PLR$Year, ASD_State_4_PLR$Prevalence)
# add SLR line
lines(ASD_State_4_PLR$Year, predict(fit_model_SLR, ASD_State_4_PLR), col="blue"
# add PLR line
lines(ASD_State_4_PLR$Year, predict(fit_model_PLR, ASD_State_4_PLR), col="orange")
```



PLR Prediction

```
In [78]: newdata = ASD_State_4_PLR[1,] # Copy datastructure
newdata$Prevalence = NA
newdata$Year = 2025

newdata
```

Prevalence	Year
849	NA 2025

```
In [79]: predict(fit_model_PLR, newdata, interval = "predict")
#
cat("Predicted ASD Prevalence of Year [", newdata$Year, "] is", round(predict(
```

fit	lwr	upr	
849	25.42036	24.34469	26.49602

Predicted ASD Prevalence of Year [2025] is 25.4 per 1,000 Children

Multiple PLR Workshop Task: 5. e. Multiple polynomial regression (MPR). (Enhance MLR by adding higher order transformed variables.)

Resuse MLR data: [ASD_State_4_MLR](#) Cop to new dataframe: ASD_State_4_MPR

```
In [80]: ASD_State_4_MPR = ASD_State_4_MLR
```

```
dim(ASD_State_4_MPR)
```

```
1692 5
```

```
In [81]: summary(ASD_State_4_MPR)
```

Denominator	Prevalence	Year	Source
Min. : 965	Min. : 0.400	Min. :2000	addm: 86
1st Qu.: 107151	1st Qu.: 3.100	1st Qu.:2003	medi:655
Median : 353328	Median : 5.600	Median :2007	nsch: 98
Mean : 604689	Mean : 7.191	Mean :2007	sped:853
3rd Qu.: 767928	3rd Qu.: 9.200	3rd Qu.:2011	
Max. :5824922	Max. :42.700	Max. :2016	

State_Full2
AZ-Arizona : 40
MD-Maryland : 40
GA-Georgia : 39
MO-Missouri : 39
NC-North Carolina: 39
WI-Wisconsin : 39
(Other) :1456

```
In [82]: # Check whether each columns got missing value:
```

```
lapply(ASD_State_4_MLR, function(col_x)sum(is.na(col_x)))
```

```
# Adjust in-line plot size to M x N
```

```
options(repr.plot.width=8, repr.plot.height=3)
```

```
barplot(apply(ASD_State_4_MLR, 2, function(col_x)sum(is.na(col_x))))
```

\$Denominator

0

\$Prevalence

0

\$Year

0

\$Source

0

\$State_Full2

0

O
1
i
[

Build Multiple PLR model: + I(Year^2) + I(log(Denominator))

```
In [83]: fit_model_MPR = lm(Prevalence ~ . + I(Year^2) + I(log(Denominator)), data = ASD)
# summary(fit_model_MPR)
```

Call:
`lm(formula = Prevalence ~ . + I(Year^2) + I(log(Denominator)),
 data = ASD_State_4_MPR)`

Residuals:

Min	1Q	Median	3Q	Max
-9.697	-1.238	0.007	1.157	19.672

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	7.908e+04	1.309e+04	6.044	1.86e-09 *
Denominator	2.074e-06	2.340e-07	8.863	< 2e-16 *
Year	-7.943e+01	1.304e+01	-6.093	1.38e-09 *
Sourcemedi	8.189e-01	6.860e-01	1.194	0.232812

$$\text{Adjusted } R^2 = 0.7884$$

Mutiple PLR Prediciton

```
In [84]: # Copy datastructure
newdata = subset(ASD_State_4_MPR, ASD_State_4_MPR$Year == 2016 &
                 ASD_State_4_MPR$State_Full2 == 'FL-Florida' &
                 ASD_State_4_MPR$Source == 'sped')
newdata
```

Denominator	Prevalence	Year	Source	State_Full2	
1652	2555399	12.1	2016	sped	FL-Florida

```
In [85]: newdata = ASD_State_4_MPR[1,] # Copy datastructure
newdata$Prevalence = NA
newdata$Denominator = 2600000
newdata$Year = 2025
newdata$Source = "sped"
newdata$State_Full2 = "FL-Florida"

newdata
```

Denominator	Prevalence	Year	Source	State_Full2
2600000	NA	2025	sped	FL-Florida

```
In [86]: predict(fit_model_MPR, newdata, interval = "predict")
#
cat("Predicted ASD Prevalence of Year [", newdata$Year, "] is", round(predict(
```

fit	lwr	upr
22.70557	17.01847	28.39267

Predicted ASD Prevalence of Year [2025] is 22.7 per 1,000 Children

Quiz:

Compare predicted ASD prevalence and model R^2 between: Multiple Linear Regression and Multiple Polynomial Rregression.

Which prediction result would you use? Provide your justifications.

```
In [87]: # Write your code below and press Shift+Enter to execute
```

Double-click **here** for the solution.

0

Linear Model: Logistic Regression (LR)

Linear Model: Logistic Regression (LR) - Workshop Task

Workshop Task:

1. a. Get the data.
2. b. Logistic regression - Binary Class.
3. c. Logistic regression - Multi-Class.

LR Workshop Task: [1. a. Get the data.](#)

Use Case Data: ["../dataset/ADV_ASD_State_R.csv"](#)

Read in CSV data, storing as R [dataframe](#)

```
In [88]: # Read back in above saved file:  
# ASD_State <- read.csv("../dataset/ADV_ASD_State_R.csv")  
# ASD_State$Year_Factor <- factor(ASD_State$Year_Factor, ordered = TRUE) # Con  
# ASD_State$Prevalence_Risk2 = factor(ASD_State$Prevalence_Risk2, ordered=TRUE  
# ASD_State$Prevalence_Risk4 = factor(ASD_State$Prevalence_Risk4, ordered=TRUE
```

```
In [89]: head(ASD_State)
```

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_F
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizo
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Geo
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	Marylan

```
In [90]: Column_Names = c("Prevalence_Risk2", "Denominator", "Year", "Source", "State_F"  
ASD_State_4_LR_Risk2 <- ASD_State[ , (names(ASD_State) %in% Column_Names)]  
dim(ASD_State_4_LR_Risk2)  
head(ASD_State_4_LR_Risk2)
```

1692 5

Denominator	Year	Source	State_Full2	Prevalence_Risk2
45322	2000	addm	AZ-Arizona	High
43593	2000	addm	GA-Georgia	High
21532	2000	addm	MD-Maryland	High
29714	2000	addm	NJ-New Jersey	High
24535	2000	addm	SC-South Carolina	High
23065	2000	addm	WV-West Virginia	Low

```
In [91]: Column_Names = c("Prevalence_Risk4", "Denominator", "Year", "Source", "State_F  
ASD_State_4_LR_Risk4 <- ASD_State[ , (names(ASD_State) %in% Column_Names)]  
dim(ASD_State_4_LR_Risk4)  
head(ASD_State_4_LR_Risk4)
```

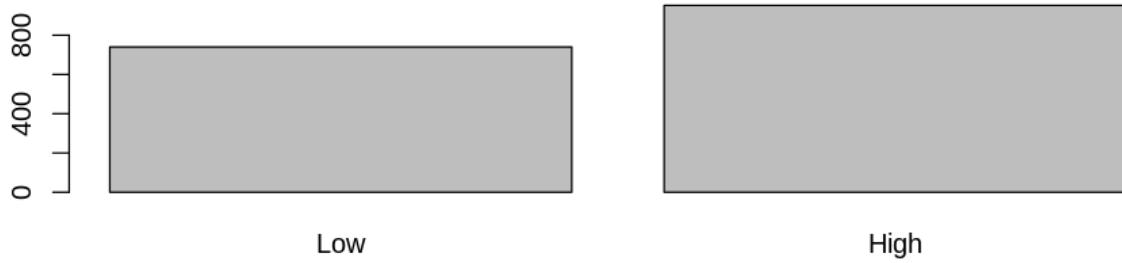
1692 5

Denominator	Year	Source	State_Full2	Prevalence_Risk4
45322	2000	addm	AZ-Arizona	Medium
43593	2000	addm	GA-Georgia	Medium
21532	2000	addm	MD-Maryland	Medium
29714	2000	addm	NJ-New Jersey	Medium
24535	2000	addm	SC-South Carolina	Medium
23065	2000	addm	WV-West Virginia	Low

LR Workshop Task: 2. b. Logistic regression (LR) Binary Class. (Reuse Multiple Polynomial Model on categorical dependent variable.)

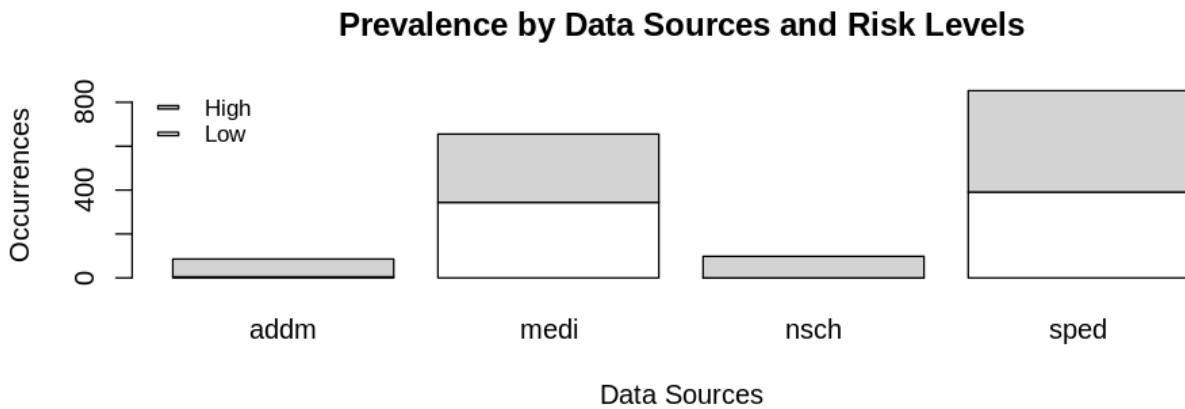
```
In [92]: table(ASD_State_4_LR_Risk2$Prevalence_Risk2)  
barplot(table(ASD_State_4_LR_Risk2$Prevalence_Risk2))
```

Low High
740 952



```
In [93]: counts = table(ASD_State_4_LR_Risk2$Prevalence_Risk2, ASD_State_4_LR_Risk2$Source)
counts
barplot(counts,
         main="Prevalence by Data Sources and Risk Levels",
         xlab="Data Sources",
         ylab="Occurrences",
         col=c("white", "lightgrey"),
         legend = rownames(counts),
         args.legend = list(x = "topleft", bty = "n", cex = 0.85, y.intersp = 4))
```

	addm	medi	nsch	sped
Low	5	344	0	391
High	81	311	98	462



```
In [94]: str(ASD_State_4_LR_Risk2)
```

```
'data.frame': 1692 obs. of 5 variables:
 $ Denominator    : int 45322 43593 21532 29714 24535 23065 35472 45113 364
72 11020 ...
 $ Year           : int 2000 2000 2000 2000 2000 2000 2002 2002 2002
...
 $ Source          : Factor w/ 4 levels "addm","medi",...
32 41 50 2 4 3 6 ...
 $ State_Full2     : Factor w/ 51 levels "AK-Alaska","AL-Alabama",...
 $ Prevalence_Risk2: Ord.factor w/ 2 levels "Low"  
"High": 2 2 2 2 2 1 1 2 2 2
...
```

Build model

```
In [95]: # Binary Classification:  
fit_model_LR_Risk2 = glm(Prevalence_Risk2 ~ Denominator + Year + Source + Stat  
                           family=binomial(link='logit'), data = ASD_State_4_LR_  
#  
summary(fit_model_LR_Risk2)  
  
Warning message:  
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
```

Evaluate model

```
In [96]: # Likelihood ratio test: significance of the difference between the full model  
pchisq(fit_model_LR_Risk2>null.deviance - fit_model_LR_Risk2$deviance,  
       fit_model_LR_Risk2$df.null - fit_model_LR_Risk2$df.residual, lower.tail
```

1.53354721010441e-271

Check whether above value is very small (the smaller the more significant), e.g. < 0.05.

< How to perform a Logistic Regression in R > Michy Alice

<https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/> (<https://www.r-bloggers.com/how-to-perform-a-logistic-regression-in-r/>)

```
In [97]: # null deviance and the residual deviance  
anova(fit_model_LR_Risk2, test="Chisq")
```

Warning message:

"glm.fit: fitted probabilities numerically 0 or 1 occurred" "glm.fit: fitted probabilities numerically 0 or 1 occurred"

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
NULL	NA	NA	1691	2318.9775	NA
Denominator	1	3.537393	1690	2315.4401	5.999971e-02
Year	1	799.973390	1689	1515.4667	5.468229e-176
Source	3	130.380991	1686	1385.0857	4.476806e-28
State_Full2	50	503.146385	1636	881.9393	4.071306e-76
I(Year^2)	1	3.124751	1635	878.8146	7.711166e-02
I(log(Denominator))	1	37.914698	1634	840.8999	7.390635e-10

```
In [98]: # R^2 equivalent  
if(!require(pscl)){install.packages("pscl")}  
library("pscl")
```

Loading required package: pscl
Classes and Methods for R developed in the
Political Science Computational Laboratory
Department of Political Science
Stanford University
Simon Jackman
hurdle and zeroinfl functions by Achim Zeileis

While no exact equivalent to the R^2 of linear regression exists, the **McFadden** R^2 index can be used to assess the model fit.

```
In [99]: # R^2 equivalent  
pR2(fit_model_LR_Risk2)[4]
```

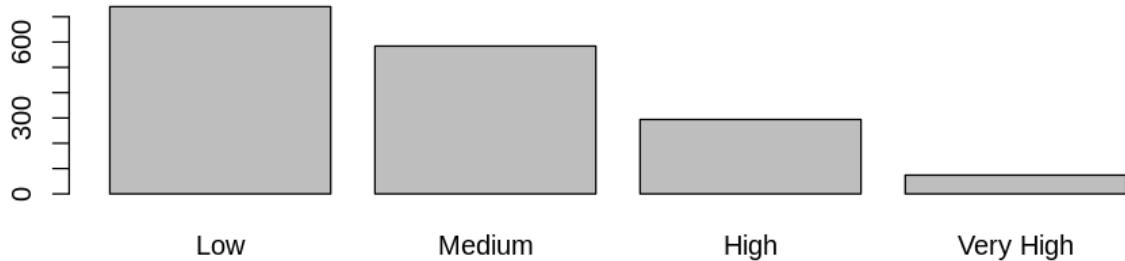
McFadden: 0.637383340253751

McFadden $R^2 = 0.6374$

LR Workshop Task: 3. c. Logistic regression (LR) Muti-Class. (Reuse Multiple Polynomial Model on categorical dependent variable.)

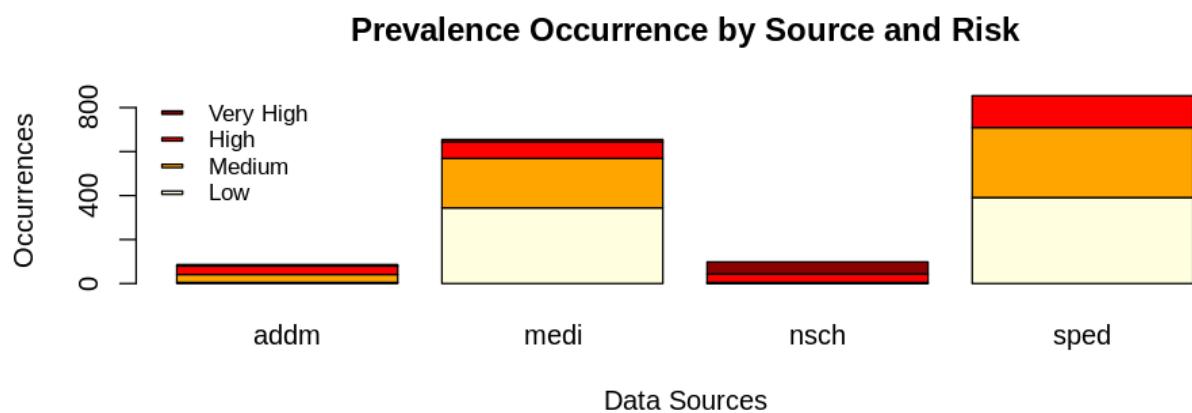
```
In [100]: table(ASD_State_4_LR_Risk4$Prevalence_Risk4)
barplot(table(ASD_State_4_LR_Risk4$Prevalence_Risk4))
```

Low	Medium	High	Very High
740	584	294	74



```
In [101]: counts = table(ASD_State_4_LR_Risk4$Prevalence_Risk4, ASD_State_4_LR_Risk4$Source)
counts
barplot(counts,
        main="Prevalence Occurrence by Source and Risk",
        xlab="Data Sources",
        ylab="Occurrences",
        col=c("lightyellow", "orange", "red", "darkred"),
        legend = rownames(counts),
        args.legend = list(x = "topleft", bty = "n", cex = 0.85, y.intersp = 4)
```

	addm	medi	nsch	sped
Low	5	344	0	391
Medium	36	225	5	318
High	38	74	38	144
Very High	7	12	55	0



Build model

```
In [102]: # multinom function from the nnet package
if(!require(nnet)){install.packages("nnet")}
library("nnet")
```

Loading required package: nnet

```
In [103]: # Multi-Class Classification:
fit_model_LR_Risk4 = multinom(Prevalence_Risk4 ~ Denominator + Year + Source +
                                data = ASD_State_4_LR_Risk4, maxit=1000) # maxit https://

summary(fit_model_LR_Risk4)

# weights: 236 (174 variable)
initial value 2345.610059
iter 10 value 1642.176333
iter 20 value 1287.779160
iter 30 value 1107.430631
iter 40 value 921.386084
iter 50 value 890.347795
iter 60 value 812.517810
iter 70 value 808.708251
iter 80 value 790.145541
iter 90 value 789.356983
iter 90 value 789.356983
iter 100 value 788.697764
iter 110 value 788.263896
iter 120 value 787.147810
iter 120 value 787.147808
iter 130 value 785.704645
iter 130 value 785.704644
iter 140 value 785.135020
... 140 value 785.135020
```

< MULTINOMIAL LOGISTIC REGRESSION | R DATA ANALYSIS EXAMPLES >

<https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/>
[\(https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/\)](https://stats.idre.ucla.edu/r/dae/multinomial-logistic-regression/)

```
In [104]: ## extract the coefficients from the model and exponentiate
exp(coef(fit_model_LR_Risk4))
# Uncomment below to display all p values:
# paste(exp(coef(fit_model)))
```

	(Intercept)	Denominator	Year	Sourcemedi	Sourcensch	Sourcesped	State_Full2AL-Alabama	Stat
Medium	0.9995993	1.000002	0.5391502	0.67139922	3.62644214	0.563177522		1.296484
High	0.9987223	1.000004	0.3065507	0.07997104	1.29768902	0.015604143		6.007789
Very High	0.9988990	1.000005	0.1954550	11.39221703	0.02789111	0.001129263		785.348612

```
In [105]: # Test the significance/importance of each coefficient (check if p values < 0.

# z score for coefficients
z <- summary(fit_model_LR_Risk4)$coefficients/summary(fit_model_LR_Risk4)$std
cat('\n< Table of coefficient z scores>')
z

# p value of 2-tailed z test
p <- (1 - pnorm(abs(z), 0, 1)) * 2
cat('\n< Table of coefficient p values>')
p
# Uncomment below to display all p values:
# paste(p)
```

< Table of coefficient z scores>

	(Intercept)	Denominator	Year	Sourcemedi	Sourcensch	Sourcesped	State_
Medium	-61605204332	22.77665	-47431585791	-3.574096e+11	5.516405e+12	-9.518100e+11	5.631
High	-103347504827	29.75500	-56434165146	-1.080017e+12	9.680944e+10	-8.591119e+12	3.547
Very High	-70795238650	2.72952	-50871679968	7.376106e+11	-1.223631e+12	-9.905923e+13	6.225

< Table of coefficient p values>

	(Intercept)	Denominator	Year	Sourcemedi	Sourcensch	Sourcesped	State_Full2AL-Alabama	State_Ful_Arka
Medium	0	0.000000000	0	0	0	0	0	0
High	0	0.000000000	0	0	0	0	0	0
Very High	0	0.006342669	0	0	0	0	0	0

While no exact equivalent to the R^2 of linear regression exists, the **McFadden** R^2 index can be used to assess the model fit.

```
In [106]: # R^2 equivalent
pR2(fit_model_LR_Risk4)[4]
```

```
fitting null model for pseudo-r2
# weights: 8 (3 variable)
initial value 2345.610059
iter 10 value 1979.347988
final value 1979.347206
converged
```

McFadden: 0.603531528984306

McFadden $R^2 = 0.6122$

Linear Model: Model Evaluation

Linear Model: Linear Model: Model Evaluation - Workshop Task

Workshop Task:

1. a. Train/Test Dataset Split
2. b. Confusion Matrix & Accuracy for Classification
3. c. K-Fold Cross Validation
4. d. R^2 , MSE, RMSE for Regression for Regression

Model Evaluation Workshop Task: 1. a. Train/Test Dataset Split

```
In [107]: if(!require(caTools)){install.packages("caTools")}  
library("caTools")
```

Loading required package: caTools

```
In [108]: # Generate a random number sequence that can be reproduced to check results th  
set.seed(88)  
  
# Stratified Random Sampling: split dataset into two sets in predefined propor  
# while preserving differnt class ratios of dependent variable. (e.g. Propor  
split <- sample.split(ASD_State_4_LR_Risk2$Prevalence_Risk2, SplitRatio = 0.7)
```

```
In [109]: # Get training and test data  
trainset <- subset(ASD_State_4_LR_Risk2, split == TRUE)  
testset <- subset(ASD_State_4_LR_Risk2, split == FALSE)
```

Build a binary classification model to predict (categorical) Prevalence Risk Level using Logistic Regression (LR)

```
In [110]: # Binary Classification:  
fit_model_LR_Risk2 = glm(Prevalence_Risk2 ~ Denominator + Year + Source + Stat  
                           family=binomial(link='logit'), data = trainset) # dat  
  
summary(fit_model_LR_Risk2)  
  
Warning message:  
“glm.fit: fitted probabilities numerically 0 or 1 occurred”
```

Model Evaluation Workshop Task: 2. b. Confusion Matrix & Accuracy for Classification

https://en.wikipedia.org/wiki/Confusion_matrix (https://en.wikipedia.org/wiki/Confusion_matrix)

```
In [111]: # Confusion matrix on Trainset  
probTrainset <- predict(fit_model_LR_Risk2, type = 'response')  
# One way is to use the proportion of High risk in the *Training* data.  
threshold2 <- sum(trainset$Prevalence_Risk2 == "High")/length(trainset$Prevalence_Risk2)  
cat('Trainset High Risk Threshold = ', threshold2)  
# If logistic regression probability > threshold, predict High, else predict Low  
predictTrainset <- ifelse(probTrainset > threshold2, "High", "Low")  
# Create a contingency table (Confusion Matrix) with actuals on rows and predicted values on columns  
table(trainset$Prevalence_Risk2, predictTrainset)  
  
# Accuracy on Trainset  
AccuracyTrain <- mean(predictTrainset == trainset$Prevalence_Risk2)  
cat('Trainset Accuracy = ', AccuracyTrain)
```

Trainset High Risk Threshold = 0.5625

	predictTrainset	
	High	Low
Low	46	472
High	593	73

Trainset Accuracy = 0.8994932

```
In [112]: # Confusion matrix on Testset
probTestset <- predict(fit_model_LR_Risk2, newdata = testset, type = 'response'
# One way is to use the proportion of High risk in the *Training* data.
cat('Reused Trainset High Risk Threshold = ', threshold2)
# If logistic regression probability > threshold, predict High, else predict L
predictTestset <- ifelse(probTestset > threshold2, "High", "Low")
# Create a contingency table (Confusion Matrix) with actuals on rows and predi
table(testset$Prevalence_Risk2, predictTestset)

# Accuracy on Trainset
AccuracyTest <- mean(predictTestset == testset$Prevalence_Risk2)
cat('Testset Accuracy = ', AccuracyTest)
```

Reused Trainset High Risk Threshold = 0.5625

		predictTestset	
		High	Low
Low	30	192	
High	258	28	

Testset Accuracy = 0.8858268

Model Evaluation Workshop Task: 3. c. K-Fold Cross Validation

R caret package

The caret package (short for Classification And REgression Training) is a set of functions that attempt to streamline the process for creating predictive models. The package contains tools for:

- data splitting
- pre-processing
- feature selection
- model tuning using resampling
- variable importance estimation

<http://topepo.github.io/caret/index.html> (<http://topepo.github.io/caret/index.html>)

```
In [113]: if(!require(caret)){install.packages("caret")}
library("caret")
```

```
Loading required package: caret
Loading required package: lattice
Loading required package: ggplot2
Registered S3 methods overwritten by 'ggplot2':
  method      from
  [.quosures   rlang
  c.quosures   rlang
  print.quosures rlang
```

```
In [114]: if(!require(e1071)){install.packages("e1071")}
library("e1071")
```

Loading required package: e1071

```
In [115]: # Caret train/test split method:  
set.seed(88)  
caret_idx = createDataPartition(ASD_State_4_LR_Risk2$Prevalence_Risk2, p = 0.8  
caret_trainset = ASD_State_4_LR_Risk2[caret_idx, ]  
caret_testset = ASD_State_4_LR_Risk2[-caret_idx, ]  
  
#caret_threshold <- sum(caret_trainset$Prevalence_Risk2 == "Low")/length(caret_trainset)  
#caret_threshold
```

```
In [116]: # Caret logistic regression  
set.seed(88)  
cv_control = trainControl(method = "cv", number = 5)  
  
caret_model_LR_Risk2 = train(form = Prevalence_Risk2 ~ ., data = caret_trainset,  
                             trControl = cv_control, method = "glm", family =  
                             "#  
                             caret_model_LR_Risk2
```

Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"

Generalized Linear Model

1439 samples
4 predictor
2 classes: 'Low', 'High'

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 1151, 1151, 1151, 1151, 1152

Resampling results:

Accuracy	Kappa
0.8638018	0.7237422

```
In [117]: # Get predicted class label  
caret_model_LR_Risk2_Pred <- predict(caret_model_LR_Risk2, caret_testset)  
  
# Uncomment below to get class probability  
# caret_model_LR_Risk2_Pred <- predict(caret_model_LR_Risk2, caret_testset, type = "prob")
```

<https://topepo.github.io/caret/using-your-own-model-in-train.html#Illustration5>
(<https://topepo.github.io/caret/using-your-own-model-in-train.html#Illustration5>).

```
In [118]: # CM & Acc
cm_table <- table(as.factor(caret_model_LR_Risk2_Pred), caret_testset$Prevalence)
confusionMatrix(cm_table)
```

Confusion Matrix and Statistics

	Low	High
Low	92	11
High	19	131

Accuracy : 0.8814
95% CI : (0.8351, 0.9185)
No Information Rate : 0.5613
P-Value [Acc > NIR] : <2e-16

Kappa : 0.7573

McNemar's Test P-Value : 0.2012

Sensitivity : 0.8288
Specificity : 0.9225
Pos Pred Value : 0.8932
Neg Pred Value : 0.8733
Prevalence : 0.4387
Detection Rate : 0.3636
Detection Prevalence : 0.4071
Balanced Accuracy : 0.8757

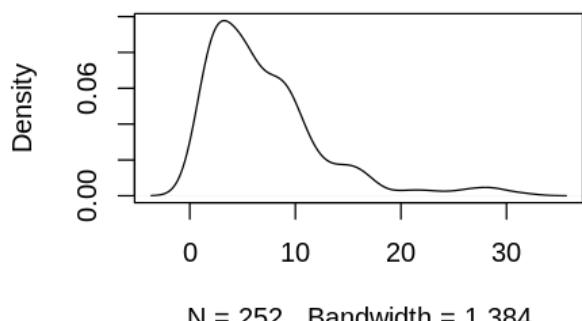
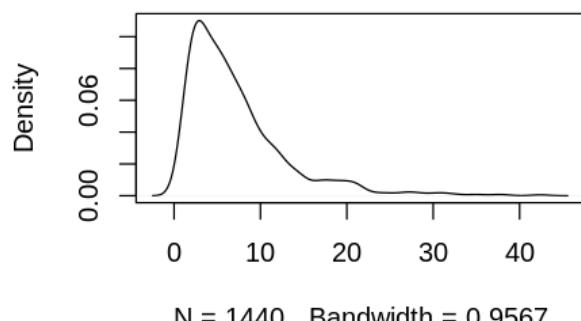
'Positive' Class : Low

Model Evaluation Workshop Task: 4. d. R^2 , MSE, RMSE for Regression

```
In [119]: # Caret train/test split method:
set.seed(88)
caret_idx = createDataPartition(ASD_State_4_MLR$Prevalence, p = 0.85, list = FALSE)
caret_trainset = ASD_State_4_MLR[caret_idx, ]
caret_testset = ASD_State_4_MLR[-caret_idx, ]
```

```
In [120]: # Look at below plots of train and test, shape of distribution are similar, which indicates good model fit
par(mfrow=c(1, 2))
plot(density(caret_trainset$Prevalence))
plot(density(caret_testset$Prevalence))
par(mfrow=c(1, 1))
```

`density.default(x = caret_trainset$Prevalence)` `density.default(x = caret_testset$Prevalence)`



(MLR)

```
In [121]: if(!require(elasticnet)){install.packages("elasticnet")}
library("elasticnet")
```

Loading required package: elasticnet
Loading required package: lars
Loaded lars 1.2

```
In [122]: # Caret MLR regression
```

```
set.seed(88)
cv_control = trainControl(method = "cv", number = 5)
#
caret_model_MLR <- train(Prevalence ~ ., data = caret_trainset, trControl = cv
#
caret_model_MLR
```

Linear Regression

1440 samples
4 predictor

No pre-processing

Resampling: Cross-Validated (5 fold)

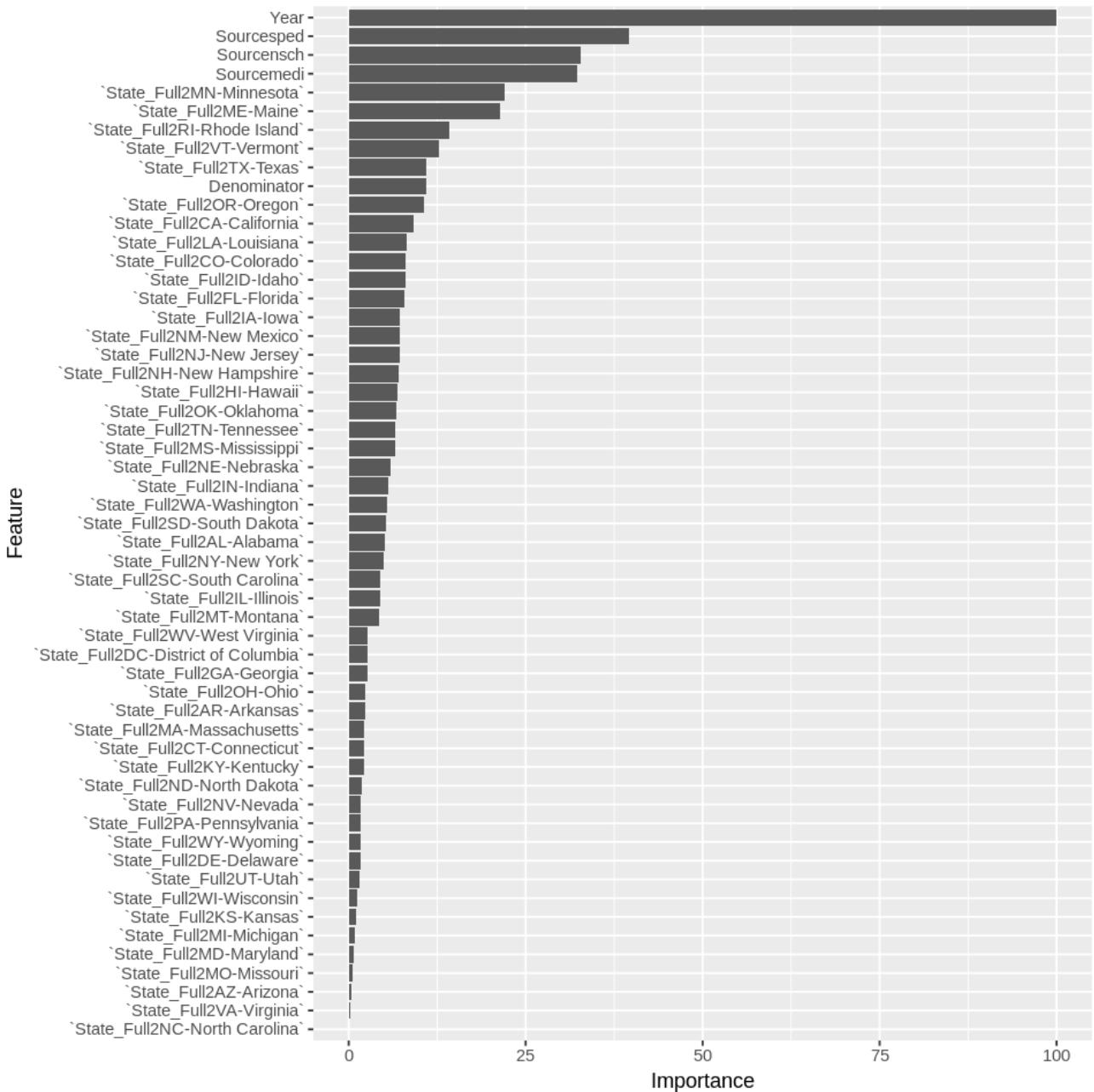
Summary of sample sizes: 1151, 1152, 1151, 1154, 1152

Resampling results:

RMSE	Rsquared	MAE
2.974216	0.7446885	1.980013

Tuning parameter 'intercept' was held constant at a value of TRUE

```
In [123]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=8)
ggplot(varImp(caret_model_MLR))
# plot(varImp(caret_model_MLR))
```



```
In [124]: caret_model_MLR_Pred <- predict(caret_model_MLR, caret_testset)
head(caret_model_MLR_Pred)
```

22	9.38901166070576
27	7.83151887672917
33	9.67043449997018
36	10.6096278033993
41	12.0705916087649
55	13.4026625975964

```
In [125]: # MSE
mean((caret_testset$Prevalence - caret_model_MLR_Pred)^2)
```

5.83744775233841

```
In [126]: # RMSE
sqrt(mean((caret_testset$Prevalence - caret_model_MLR_Pred)^2))

2.41608107321307
```

Enhanced MLR using Regularization: L1 Lasso

<https://towardsdatascience.com/create-predictive-models-in-r-with-caret-12baf9941236>
[\(https://towardsdatascience.com/create-predictive-models-in-r-with-caret-12baf9941236\)](https://towardsdatascience.com/create-predictive-models-in-r-with-caret-12baf9941236)

```
In [127]: if(!require(elasticnet)){install.packages("elasticnet")}
library("elasticnet")
```

```
In [128]: # Caret MLR with Regularization
# possible method: "boot", "boot632", "cv", "repeatedcv", "L00CV", "LG0CV"
cv_control <- trainControl(method = "repeatedcv",
                            number = 10,      # number of folds
                            repeats = 5)     # repeated N times

caret_model_MLR_L1 <- train(Prevalence ~ .,
                             data = caret_trainset,
                             method = "lasso", # Try using "ridge"
                             trControl = cv_control,
                             preprocess = c('scale', 'center')) # Auto pre-process data
#
caret_model_MLR_L1
```

The lasso

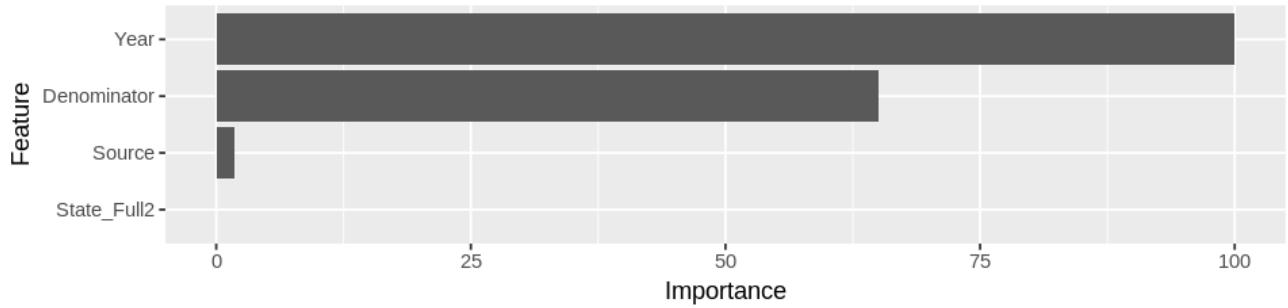
1440 samples
4 predictor

Pre-processing: scaled (55), centered (55)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 1296, 1295, 1296, 1296, 1296, 1296, ...
Resampling results across tuning parameters:

fraction	RMSE	Rsquared	MAE
0.1	4.586996	0.5968962	3.264324
0.5	3.141998	0.7206010	2.138362
0.9	2.925833	0.7546077	1.966103

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was fraction = 0.9.

```
In [129]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=2)
ggplot(varImp(caret_model_MLR_L1))
# plot(varImp(caret_model_MLR_L1))
```



```
In [130]: caret_model_MLR_Pred <- predict(caret_model_MLR_L1, caret_testset)
head(caret_model_MLR_Pred)
```

22	9.09559548122945
27	7.65758256776709
33	9.71165245354341
36	10.4202137443522
41	11.766911449798
55	13.0976251287959

```
In [131]: # MSE
mean((caret_testset$Prevalence - caret_model_MLR_Pred)^2)
```

5.90034443032887

```
In [132]: # RMSE
sqrt(mean((caret_testset$Prevalence - caret_model_MLR_Pred)^2))
```

2.42906245912469

Enhanced MLR using Regularization: L2 Ridge

```
In [133]: # Caret MLR with Regularization
# possible method: "boot", "boot632", "cv", "repeatedcv", "LOOCV", "LGOCV"
cv_control <- trainControl(method = "repeatedcv",
                            number = 10,      # number of folds
                            repeats = 5)     # repeated N times

caret_model_MLR_L2 <- train(Prevalence ~ .,
                             data = caret_trainset,
                             method = "ridge", # Try using "lasso"
                             trControl = cv_control,
                             preProcess = c('scale', 'center')) # Auto pre-process data
#
caret_model_MLR_L2
```

Ridge Regression

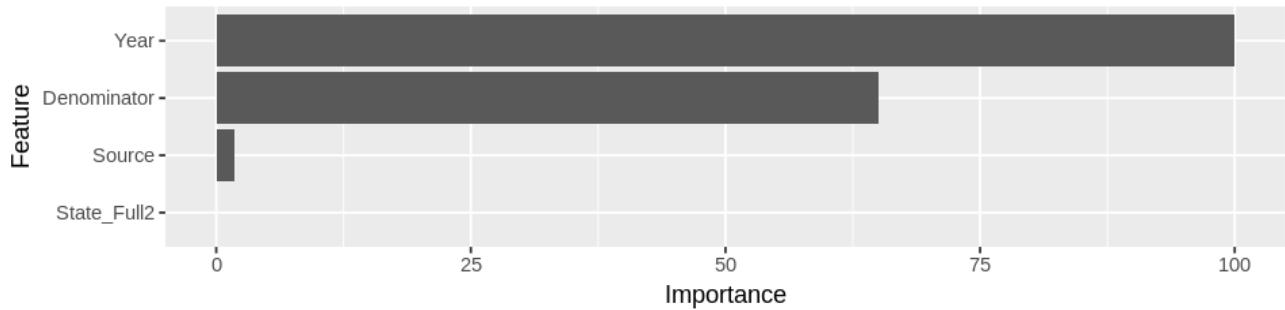
1440 samples
4 predictor

Pre-processing: scaled (55), centered (55)
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 1296, 1297, 1296, 1297, 1295, 1296, ...
Resampling results across tuning parameters:

lambda	RMSE	Rsquared	MAE
0e+00	2.951386	0.7509153	1.966418
1e-04	2.951369	0.7509168	1.966466
1e-01	3.000216	0.7427197	2.048206

RMSE was used to select the optimal model using the smallest value.
The final value used for the model was lambda = 1e-04.

```
In [134]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=2)
ggplot(varImp(caret_model_MLR_L2))
# plot(varImp(caret_model_MLR_L2))
```



```
In [135]: caret_model_MLR_Pred <- predict(caret_model_MLR_L2, caret_testset)
head(caret_model_MLR_Pred)
```

22	9.38476017497349
27	7.82653071659292
33	9.66714756932461
36	10.6063229876567
41	12.0663388797878
55	13.3984373159588

```
In [136]: # MSE  
mean((caret_testset$Prevalence - caret_model_MLR_Pred)^2)  
5.83820592478024
```

```
In [137]: # RMSE  
sqrt(mean((caret_testset$Prevalence - caret_model_MLR_Pred)^2))  
2.41623796940207
```

```
.0
```

Workshop Submission

What to submit?

Create predictive model for Multi Class Classification of ASD Prevalence Risk Level (Low, Medium, High, Very High) using Caret package's multinom logistic regression algorithm.

References:

[\(https://daviddalpiaz.github.io/r4sl/the-caret-package.html\)](https://daviddalpiaz.github.io/r4sl/the-caret-package.html)

[\(http://topepo.github.io/caret/index.html\)](http://topepo.github.io/caret/index.html)

[\(https://cran.r-project.org/web/packages/caret/caret.pdf\)](https://cran.r-project.org/web/packages/caret/caret.pdf)

```
In [138]: # Code example of multinomial logistic regression using Iris flower dataset
iris[c(1:3, 51:53, 101:103), ]
summary(iris)
iris_model = train(Species ~ ., data = iris, method = "multinom",
                    trControl = trainControl(method = "cv", number = 5), trace = TRUE)
iris_model
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
51	7.0	3.2	4.7	1.4	versicolor
52	6.4	3.2	4.5	1.5	versicolor
53	6.9	3.1	4.9	1.5	versicolor
101	6.3	3.3	6.0	2.5	virginica
102	5.8	2.7	5.1	1.9	virginica
103	7.1	3.0	5.9	2.1	virginica

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Min.	:4.300	:2.000	:1.000	:0.100
1st Qu.	:5.100	:2.800	:1.600	:0.300
Median	:5.800	:3.000	:4.350	:1.300
Mean	:5.843	:3.057	:3.758	:1.199
3rd Qu.	:6.400	:3.300	:5.100	:1.800
Max.	:7.900	:4.400	:6.900	:2.500

Species
setosa :50
versicolor:50
virginica :50

Penalized Multinomial Regression

```
150 samples
 4 predictor
 3 classes: 'setosa', 'versicolor', 'virginica'
```

No pre-processing

Resampling: Cross-Validated (5 fold)

Summary of sample sizes: 120, 120, 120, 120, 120

Resampling results across tuning parameters:

decay	Accuracy	Kappa
0e+00	0.9533333	0.93
1e-04	0.9600000	0.94
1e-01	0.9733333	0.96

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was decay = 0.1.

```
In [139]: # Write your code below and press Shift+Enter to execute
```

Double-click **here** for the solution.

Excellent! You have completed the workshop notebook!

Connect with the author:

This notebook was written by [GU Zhan \(Sam\)](https://sg.linkedin.com/in/zhan-gu-27a82823).

[Sam](https://www.iss.nus.edu.sg/about-us/staff/detail/201/GU%20Zhan) is currently a lecturer in [Institute of Systems Science](https://www.iss.nus.edu.sg/) in [National University of Singapore](http://www.nus.edu.sg/). He devotes himself into pedagogy & andragogy, and is very passionate in inspiring next generation of artificial intelligence lovers and leaders.

Copyright © 2020 GU Zhan

This notebook and its source code are released under the terms of the [MIT License](https://en.wikipedia.org/wiki/MIT_License).

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Appendices

Interactive workshops: < Learning R inside R > using swirl() (in R/RStudio)

<https://github.com/telescopeuser/S-SB-Workshop>

<https://github.com/dd-consulting>



閱覽室