



One-Stop Analytics: Exploratory Data Analysis (EDA) & Statistics

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]

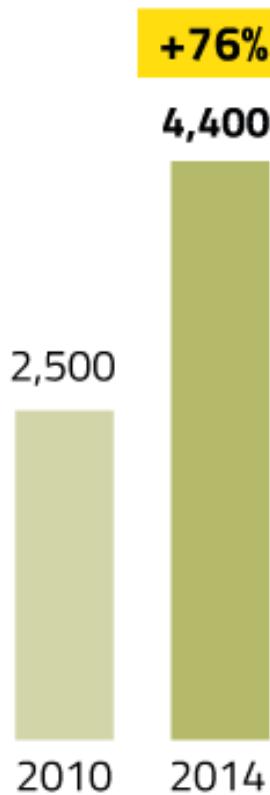
TODAY Online - More preschoolers diagnosed with developmental issues

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

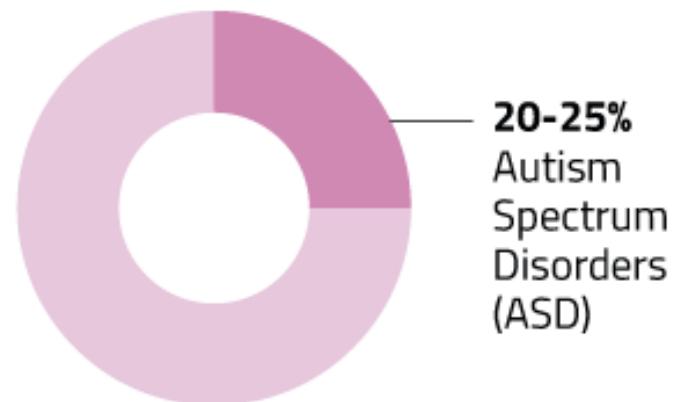
<https://www.gov.sg/news/content/today-online-more-preschoolers-diagnosed-with-developmental-issues>
<https://www.gov.sg/news/content/today-online-more-preschoolers-diagnosed-with-developmental-issues>

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


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



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

.0

Workshop Objective:

Use R to analyze Autism Spectrum Disorder (ASD) data from CDC USA.

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

- **R Fundamentals**
- **Data Summarization**
- **Data Visualisation (Base Graphic)**
- **Data Visualisation (Enhanced)**
- **Sampling & Normality**
- **Confidence Interval (CI)**
- **Workshop Submission**
- **Appendices**

.0

R Fundamentals

R Fundamentals - Get & Set working directory

Obtain current R **working directory**

In [1]: `getwd()`

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

Set new R working directory

In [2]: `# 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'
```

Read in CSV data, storing as R **dataframe**

```
In [3]: # Dataset: US. National Level Children ASD Prevalence  
ASD_National <- read.csv("../dataset/ADV_ASD_National.csv", stringsAsFactors =  
  
In [4]: # Dataset: US. State Level Children ASD Prevalence  
ASD_State     <- read.csv("../dataset/ADV_ASD_State.csv", stringsAsFactors = FA
```

Obtain number of rows and number of columns/features/variables

```
In [5]: dim(ASD_National)
```

```
42 26
```

```
In [6]: dim(ASD_State)
```

```
1692 48
```

Obtain overview (data structure/types)

```
In [7]: str(ASD_National)
```

```
'data.frame': 42 obs. of 26 variables:  
 $ Source : chr "addm" "addm" "addm" "addm"  
 ...  
 $ Year   : int 2000 2002 2004 2006 2008 2010  
 2012 2014 2004 2008 ...  
 $ Prevalence : num 6.7 6.6 8 9 11.3 14.7 14.8 1  
 6.8 9.5 16.2 ...  
 $ Upper.CI  : num 7 6.8 8.4 9.3 11.7 15.1 15.2  
 17.3 12 18.1 ...  
 $ Lower.CI  : num 6.3 6.3 7.6 8.6 11 14.3 14.4  
 16.4 7.4 14.5 ...  
 $ Prevalence_dup : num 6.7 6.6 8 9 11.3 14.7 14.8 1  
 6.8 9.5 16.2 ...  
 $ Source_Full1 : chr "Autism & Developmental Disab  
 ilities Monitoring Network" "Autism & Developmental Disabilities Monitoring  
 Network" "Autism & Developmental Disabilities Monitoring Network" "Autism &  
 Developmental Disabilities Monitoring Network" ...  
 $ Source_Full2 : chr "addm-Autism & Developmental  
 Disabilities Monitoring Network" "addm-Autism & Developmental Disabilities  
 Monitoring Network" "addm Autism & Developmental Disabilities Monitoring No
```

In [8]: str(ASD_State)

```
'data.frame': 1692 obs. of 48 variables:
 $ State                               : chr  "AZ" "GA" "MD" "NJ" ...
 $ Denominator                         : int  45322 43593 21532 29714 245
 $ 35 23065 35472 45113 36472 11020 ...
 $ Prevalence                           : num  6.5 6.5 5.5 9.9 6.3 4.5 3.3
 $ 6.2 6.9 5.9 ...
 $ Lower.CI                            : num  5.8 5.8 4.6 8.9 5.4 3.7 2.7
 $ 5.5 6.1 4.6 ...
 $ Upper.CI                            : num  7.3 7.3 6.6 11.1 7.4 5.5 3.
 $ 9 7 7.8 7.5 ...
 $ Year                                 : int  2000 2000 2000 2000 2000 2000 20
 $ 00 2002 2002 2002 2002 ...
 $ Source                               : chr  "addm" "addm" "addm" "addm"
 ...
 $ Source_Full1                         : chr  "Autism & Developmental Dis-
 abilities Monitoring Network" "Autism & Developmental Disabilities Monitori-
 ng Network" "Autism & Developmental Disabilities Monitoring Network" "Autis-
 m & Developmental Disabilities Monitoring Network" ...
 $ Source_Full1                         : chr  "Arizona" "Georgia" "Maryla
 ...
 "Maryland" ...
```

Look at first few rows of data

In [9]: head(ASD_National)

Source	Year	Prevalence	Upper.Cl	Lower.Cl	Prevalence_dup	Source_Full1	Source_Full2	Male.Pre
addm	2000	6.7	7.0	6.3	6.7	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	
addm	2002	6.6	6.8	6.3	6.6	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	
addm	2004	8.0	8.4	7.6	8.0	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	

In [10]: head(ASD_State)

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_Full
AZ	45322	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizona
GA	43593	6.5	5.8	7.3	2000	addm	Autism & Developmental Disabilities Monitoring Network	Georgia	GA-Georgia
MD	21532	5.5	4.6	6.6	2000	addm	Autism & Developmental Disabilities Monitoring Network	Maryland	MD Maryland
NJ	29714	9.9	8.9	11.1	2000	addm	Autism & Developmental Disabilities Monitoring Network	New Jersey	NJ-New Jersey
SC	24535	6.3	5.4	7.4	2000	addm	Autism & Developmental Disabilities Monitoring Network	South Carolina	SC-Sout Carolin
WV	23065	4.5	3.7	5.5	2000	addm	Autism & Developmental Disabilities Monitoring Network	West Virginia	WV-We Virgin

Obtain name of columns

In [11]: names(ASD_National)

```
'Source'  'Year'  'Prevalence'  'Upper.CI'  'Lower.CI'  'Prevalence_dup'  'Source_Full1'  
'Source_Full2'  '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 [12]: names(ASD_State)

```
'State' 'Denominator' 'Prevalence' 'Lower.CI' 'Upper.CI' 'Year' 'Source' 'Source_Full1'  
'State_Full1' 'State_Full2' 'Numerator_ASD' 'Numerator_NonASD' 'Proportion' 'X95_Z_CI'  
'Z_Lower.CI' 'Z_Upper.CI' 'Z_Lower.CI_ABSerror' 'Z_Upper.CI_ABSerror' 'Chi_Wilson_P'  
'X95_Chi_Wilson_CI' 'Chi_Wilson_Lower.CI' 'Chi_Wilson_Upper.CI'  
'Chi_Wilson_Lower.CI_ABSerror' 'Chi_Wilson_Upper.CI_ABSerror'  
'Chi_Wilson_Corrected_w_minus.CI' 'Chi_Wilson_Corrected_w_plus.CI'  
'Chi_Wilson_Corrected_Lower.CI' 'Chi_Wilson_Corrected_Upper.CI'  
'Chi_Wilson_Corrected_Lower.CI_ABSerror' 'Chi_Wilson_Corrected_Upper.CI_ABSerror'  
'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'
```

Display column name with its index number

In [13]: cbind(names(ASD_National), c(1:length(names(ASD_National))))

Source	1
Year	2
Prevalence	3
Upper.CI	4
Lower.CI	5
Prevalence_dup	6
Source_Full1	7
Source_Full2	8
Male.Prevalence	9
Male.Lower.CI	10
Male.Upper.CI	11
Female.Prevalence	12
Female.Lower.CI	13

Look at data structure/schema (Selected columns)

```
In [14]: str(ASD_National[, c(1:8, 24, 25, 26)])
```

```
'data.frame': 42 obs. of 11 variables:  
 $ Source : chr "addm" "addm" "addm" "addm" ...  
 $ Year   : int 2000 2002 2004 2006 2008 2010 2  
 $ 012    : num 2004 2008 ...  
 $ Prevalence : num 6.7 6.6 8 9 11.3 14.7 14.8 16.8  
 $ 9.5    : num 16.2 ...  
 $ Upper.CI : num 7 6.8 8.4 9.3 11.7 15.1 15.2 1  
 $ 7.3    : num 12 18.1 ...  
 $ Lower.CI : num 6.3 6.3 7.6 8.6 11 14.3 14.4 1  
 $ 6.4    : num 7.4 14.5 ...  
 $ Prevalence_dup : num 6.7 6.6 8 9 11.3 14.7 14.8 16.8  
 $ 9.5    : num 16.2 ...  
 $ Source_Full1 : chr "Autism & Developmental Disabilities Monitoring Network" ...  
 $ Source_Full2 : chr "addm-Autism & Developmental Disabilities Monitoring Network" ...  
 $ Asian.or.Pacific.Islander.Prevalence: chr "No data" "No data" "No data"  
 "No data" ...  
 $ Asian.or.Pacific.Islander.Lower.CI : chr "No data" "No data" "No data"  
 "No data" ...  
 $ Asian.or.Pacific.Islander.Upper.CI : chr "No data" "No data" "No data"  
 "No data" ...
```

Quiz:

Obtain feature/column names and column index of dataframe: ASD_State

```
In [15]: # Write your code below and press Shift+Enter to execute
```

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

R Fundamentals - Work with dataframe

Access column 1 as a **named list** of string/chr (dataframe alike):

```
In [16]: # use column index:  
ASD_National[1]
```

Source

addm
addm
addm
addm
addm
addm
addm
addm
nsch
nsch
nsch
nsch

```
In [17]: typeof(ASD_National[1])
```

'list'

```
In [18]: ASD_National[1]$Source
```

'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'nsch' 'nsch' 'nsch'
'sped'
'sped' 'sped' 'sped' 'sped' 'sped' 'medi' 'medi' 'medi' 'medi' 'medi'
'medi' 'medi' 'medi' 'medi' 'medi' 'medi' 'medi' 'medi'

```
In [19]: # use column name:  
ASD_National["Source"]
```

Source

addm
addm
addm
addm
addm
addm
addm
addm
nsch
nsch
nsch
nsch

```
In [20]: typeof(ASD_National['Source'])
```

'list'

```
In [21]: ASD_National['Source']$Source
```

```
'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'nsch' 'nsch' 'nsch'  
'nsch' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

Access column 1 as a set of string/chr:

```
In [22]: ASD_National[, 1]
```

```
'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'nsch' 'nsch' 'nsch'  
'nsch' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

```
In [23]: typeof(ASD_National[, 1])
```

```
'character'
```

```
In [24]: ASD_National[, 1]
```

```
'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'nsch' 'nsch' 'nsch'  
'nsch' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

```
In [25]: # or  
ASD_National[, "Source"]
```

```
'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'nsch' 'nsch' 'nsch'  
'nsch' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

```
In [26]: # or  
ASD_National$Source
```

```
'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'addm' 'nsch' 'nsch' 'nsch'  
'nsch' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'  
'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

Count number of elements in a object:

```
In [27]: length(ASD_National) # number of features/columns
```

```
26
```

```
In [28]: length(ASD_National[1, ]) # number of features/columns
```

```
26
```

```
In [29]: length(ASD_National[, 1]) # number of elements(rows) in column 1
```

42

```
In [30]: length(ASD_National[, "Source"]) # same as above
```

42

```
In [31]: length(ASD_National$Source) # number of elements in chr list
```

42

Access elements from column 1, which is a list:

```
In [32]: # using column index  
ASD_National[1][1, ]
```

'addm'

```
In [33]: ASD_National[1][11, ]
```

'nsch'

```
In [34]: ASD_National[1][11:20, ]
```

'nsch' 'nsch' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'

```
In [35]: # using column name  
ASD_National["Source"][1, ]
```

'addm'

```
In [36]: ASD_National["Source"][11, ]
```

'nsch'

```
In [37]: ASD_National["Source"][11:20, ]
```

'nsch' 'nsch' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'

Access elements from column 1, which is a set of string/chr:

```
In [38]: # using column index  
ASD_National[, 1][1]
```

'addm'

```
In [39]: ASD_National[, 1][11]
```

'nsch'

```
In [40]: ASD_National[, 1][11:20]
```

'nsch' 'nsch' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'

```
In [41]: # using column name  
ASD_National[, "Source"][1]  
  
'addm'
```

```
In [42]: # using column name  
ASD_National[, "Source"][11]  
  
'nsch'
```

```
In [43]: # using column name  
ASD_National[, "Source"][11:20]  
  
'nsch' 'nsch' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

```
In [44]: # using $ operator  
ASD_National$Source[1]  
  
'addm'
```

```
In [45]: ASD_National$Source[11]  
  
'nsch'
```

```
In [46]: ASD_National$Source[11:20]  
  
'nsch' 'nsch' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped' 'sped'
```

Access elements of different column:

```
In [47]: cbind(names(ASD_National), c(1:length(names(ASD_National))))
```

Source	1
Year	2
Prevalence	3
Upper.Cl	4
Lower.Cl	5
Prevalence_dup	6
Source_Full1	7
Source_Full2	8
Male.Prevalence	9
Male.Lower.Cl	10
Male.Upper.Cl	11
Female.Prevalence	12
Female.Lower.Cl	13

```
In [48]: ASD_National[1, 1] # row 1, column 1: "Source"  
  
'addm'
```

```
In [49]: ASD_National[10, 1] # row 10, column 1: "Source"  
  
'nsch'
```

```
In [50]: ASD_National[1, 3] # row 1, column 3: "Prevalence"
```

6.7

```
In [51]: ASD_National[10, 3] # row 10, column 3: "Prevalence"
```

16.2

```
In [52]: ASD_National[1:10, 1:3] # row 1 to 10 from column 1 to 3
```

Source	Year	Prevalence
addm	2000	6.7
addm	2002	6.6
addm	2004	8.0
addm	2006	9.0
addm	2008	11.3
addm	2010	14.7
addm	2012	14.8
addm	2014	16.8
nsch	2004	9.5
nsch	2008	16.2

```
In [53]: # or using columns names
```

```
ASD_National[1:10, c('Source', 'Year', 'Prevalence')]
```

Source	Year	Prevalence
addm	2000	6.7
addm	2002	6.6
addm	2004	8.0
addm	2006	9.0
addm	2008	11.3
addm	2010	14.7
addm	2012	14.8
addm	2014	16.8
nsch	2004	9.5
nsch	2008	16.2

```
In [54]: ASD_National[c(1:10, 20, 30:35), c(1:3, 9, 12)] # row 1 to 10, 20, and 20 to 2
```

	Source	Year	Prevalence	Male.Prevalence	Female.Prevalence
1	addm	2000	6.7	No data	No data
2	addm	2002	6.6	11.5	2.7
3	addm	2004	8.0	12.9	2.9
4	addm	2006	9.0	14.5	3.2
5	addm	2008	11.3	18.4	4
6	addm	2010	14.7	23.7	5.3
7	addm	2012	14.8	23.4	5.2
8	addm	2014	16.8	26.6	6.6
9	nsch	2004	9.5		
10	nsch	2008	16.2		
20	sped	2007	5.4		
30	medi	2000	2.3		
31	medi	2001	2.6		
32	medi	2002	2.8		
33	medi	2003	3.0		
34	medi	2004	3.5		
35	medi	2005	3.9		

[Tips] We notice missing data from above.

R Fundamentals - Process missing data

Count missing values in dataframe:

```
In [55]: sum(is.na(ASD_National)) # No missing data recognised by R (NA)
```

0

```
In [56]: sum(is.na(ASD_State)) # Some missing data recognised by R (NA)
```

14454

Empty string, "No data" are not considered as missing value by R, thus we need to handle them manually.

```
In [57]: # Define several offending strings  
na_strings <- c("", "No data", "NA", "N A", "N / A", "N/A", "N/ A", "Not Avail")
```

```
In [58]: # Load required function from packages:  
if(!require(naniar)){install.packages("naniar")}  
library(naniar)  
if(!require(dplyr)){install.packages("dplyr")}  
library(dplyr)
```

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

```
In [59]: # Uncomment below to show help  
# ?replace_with_na_all # Documentation
```

Replace these defined missing/offending values to R's internal NA

```
In [60]: # "~.x" is a reserved keyword of this function:  
ASD_National = replace_with_na_all(ASD_National, condition = ~.x %in% na_string)
```

```
In [61]: # Count missing values (R's internal NA) in dataframe:  
sum(is.na(ASD_National))
```

650

R Fundamentals - Process invalid characters

Remove invalid unicode char/string: \x92

```
In [62]: ASD_National$Source_Full1[ASD_National$Source_Full1 == "National Survey of Chi  
"National Survey of Children's Health"]
```

```
In [63]: ASD_National$Source_Full2[ASD_National$Source_Full2 == "nsch-National Survey o  
"nsch-National Survey of Children's Health"]
```

R Fundamentals - Delete/Drop dataframe variable

Delete/Drop duplicate variable: Prevalence_dup

```
In [64]: drop <- c("Prevalence_dup", "Dummy Variable Name")
```

```
In [65]: ASD_National = ASD_National[, !(names(ASD_National) %in% drop)] # Recall Data
```

R Fundamentals - Create/Add dataframe variable

Create one new variable: Source_UC by converting to uppercase letters

```
In [66]: ASD_National$Source_UC <- paste(toupper(ASD_National$Source))
```

Create one new variable: Source_Full3 by combining Source and Source_Full1

```
In [67]: ASD_National$Source_Full3 <- paste(toupper(ASD_National$Source), ASD_National$
```

Create one new ordinal categorical variable: Prevalence_Rank2 ("Low", "High") by binning Prevalence

```
In [68]: # Recode Risk into category from Prevalence
```

```
# Low [0, 5)
# High [5, +oo)
```

```
ASD_National$Prevalence_Risk2[ASD_National$Prevalence < 5] = "Low"
ASD_National$Prevalence_Risk2[ASD_National$Prevalence >= 5 ] = "High"
#
head(ASD_National)
```

Warning message:

"Unknown or uninitialized column: 'Prevalence_Risk2'."

Create one new ordinal categorical variable: Prevalence_Rank4 ("Low", "Medium", "High", "Very High") by binning Prevalence

```
In [69]: # Recode Risk into category from Prevalence
```

```
# Low [0, 5)
# Medium [5, 10)
# High [10, 20)
# Very High [20, +oo)

ASD_National$Prevalence_Risk4 = "Very High"
ASD_National$Prevalence_Risk4[ASD_National$Prevalence < 20] = "High"
ASD_National$Prevalence_Risk4[ASD_National$Prevalence < 10] = "Medium"
ASD_National$Prevalence_Risk4[ASD_National$Prevalence < 5] = "Low"
#
head(ASD_National)
```

Source	Year	Prevalence	Upper.Cl	Lower.Cl	Source_Full1	Source_Full2	Male.Prevalence	Male.Low
addm	2000	6.7	7.0	6.3	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	NA	
addm	2002	6.6	6.8	6.3	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	11.5	
addm	2004	8.0	8.4	7.6	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	12.9	

R Fundamentals - Convert to correct data types

Review data structure and variable names:

```
In [70]: str(ASD_National)
cbind(names(ASD_National), c(1:length(names(ASD_National))))
```

Classes 'tbl_df', 'tbl' and 'data.frame': 42 obs. of 29 variables:
\$ Source : chr "addm" "addm" "addm" "addm"
...
\$ Year : int 2000 2002 2004 2006 2008 2010
2012 2014 2004 2008 ...
\$ Prevalence : num 6.7 6.6 8 9 11.3 14.7 14.8 1
6.8 9.5 16.2 ...
\$ Upper.CI : num 7 6.8 8.4 9.3 11.7 15.1 15.2
17.3 12 18.1 ...
\$ Lower.CI : num 6.3 6.3 7.6 8.6 11 14.3 14.4
16.4 7.4 14.5 ...
\$ Source_Full1 : chr "Autism & Developmental Disabilities Monitoring Network" ...
\$ Source_Full2 : chr "addm-Autism & Developmental Disabilities Monitoring Network" ...
...

Convert Prevalence and CIs from categorical/chr to numeric, column 8 to 25

```
In [71]: ix <- 8:25 # define an index
# apply()
ASD_National[ix] <- apply(ASD_National[ix], 2, as.numeric) # "2" means column-wise
```

```
In [72]: # Uncomment below to show help
# ?apply # Documentation
```

```
In [73]: # or lapply()
ASD_National[ix] <- lapply(ASD_National[ix], as.numeric) # column-wise
```

```
In [74]: # Uncomment below to show help
# ?lapply # Documentation
```

Convert Source from categorical/chr to categorical/factor

```
In [75]: ix <- c(1, 6, 7, 26, 27) # define an index
ASD_National[ix] <- lapply(ASD_National[ix], as.factor)
```

Create new ordered factor Year_Factor from Year

```
In [76]: ASD_National$Year_Factor <- factor(ASD_National$Year, ordered = TRUE)
```

```
In [77]: # Observe the difference of 'Levels' in below two factors  
ASD_National$Year_Factor # Ordinal categorical variable  
str(ASD_National$Year_Factor)  
  
ASD_National$Source # Nominal categorical variable  
str(ASD_National$Source)
```

2000	2002	2004	2006	2008	2010	2012	2014	2004	2008	2012	2016	2000	2001
2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
2016	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012

► Levels:

```
Ord.factor w/ 17 levels "2000"<"2001"<...: 1 3 5 7 9 11 13 15 5 9 ...  
  
addm addm addm addm addm addm nsch nsch nsch sped sped  
sped sped sped sped sped sped sped sped sped sped sped  
sped sped medi medi medi medi medi medi medi medi medi medi
```

► Levels:

```
Factor w/ 4 levels "addm", "medi", ...: 1 1 1 1 1 1 1 1 3 3 ...
```

Convert Prevalence_Rank2 & Prevalence_Rank4 to ordered factor

```
In [78]: # Convert to factor  
ASD_National$Prevalence_Risk2 = factor(ASD_National$Prevalence_Risk2, ordered=TRUE,  
                                         levels=c("Low", "High"))  
  
# Convert to factor  
ASD_National$Prevalence_Risk4 = factor(ASD_National$Prevalence_Risk4, ordered=TRUE,  
                                         levels=c("Low", "Medium", "High", "Very High"))
```

```
In [79]: # Optionally, below is manual conversion examples:  
# ASD_National$Male.Prevalence = as.numeric(ASD_National$Male.Prevalence)  
# ASD_National$Source = as.factor(ASD_National$Source)  
# ASD_National$Prevalence_Risk2 = factor(ASD_National$Prevalence_Risk2, ordered=TRUE)  
# ASD_National$Prevalence_Risk4 = factor(ASD_National$Prevalence_Risk4, ordered=TRUE)
```

Optionally, export the processed dataframe data to CSV file.

```
In [80]: write.csv(ASD_National, file = "../dataset/ADV_ASD_National_R.csv", row.names=FALSE)
```

```
In [81]: # Read back in above saved file:  
# ASD_National <- read.csv("../dataset/ADV_ASD_National_R.csv")  
# ASD_National$Year_Factor <- factor(ASD_National$Year_Factor, ordered = TRUE)
```

Data Summarization

Data Summarization - High Level Data Summary

```
In [82]: summary(ASD_National)
```

Source	Year	Prevalence	Upper.CI	Lower.CI
addm: 8	Min. :2000	Min. : 1.800	Min. : 1.800	Min. : 1.700
medi:13	1st Qu.:2004	1st Qu.: 3.950	1st Qu.: 3.950	1st Qu.: 3.875
nsch: 4	Median :2008	Median : 6.650	Median : 6.900	Median : 6.350
sped:17	Mean :2007	Mean : 7.952	Mean : 8.207	Mean : 7.712
	3rd Qu.:2011	3rd Qu.: 9.725	3rd Qu.:10.350	3rd Qu.: 9.625
	Max. :2016	Max. :29.200	Max. :30.700	Max. :27.700

Source_Full1
Autism & Developmental Disabilities Monitoring Network: 8
Medicaid :13
National Survey of Children's Health : 4

Data Summarization - Summary of numeric variables

```
In [83]: # Filter only numeric variables/columns  
select_if(ASD_National, is.numeric) # library(dplyr)
```

Year	Prevalence	Upper.CI	Lower.CI	Male.Prevalence	Male.Lower.CI	Male.Upper.CI	Female.Prevalence	Female.Lower.CI	Female.Upper.CI
2000	6.7	7.0	6.3	NA	NA	NA	NA	NA	NA
2002	6.6	6.8	6.3	11.5	NA	NA	NA	NA	2.1
2004	8.0	8.4	7.6	12.9	12.2	13.7	2.1	NA	NA
2006	9.0	9.3	8.6	14.5	13.9	15.1	3.1	NA	NA
2008	11.3	11.7	11.0	18.4	17.7	19.0	4.1	NA	NA
2010	14.7	15.1	14.3	23.7	23.0	24.4	5.1	NA	NA
2012	14.8	15.2	14.4	23.4	22.7	24.1	5.1	NA	NA
2014	16.8	17.3	16.4	26.6	25.8	27.4	6.1	NA	NA
2004	9.5	12.0	7.4	NA	NA	NA	NA	NA	NA
2008	16.2	18.1	14.5	NA	NA	NA	NA	NA	NA
2012	21.2	22.3	20.1	NA	NA	NA	NA	NA	NA

```
In [84]: # Data summarization  
summary(select_if(ASD_National, is.numeric))
```

Year	Prevalence	Upper.CI	Lower.CI
Min. :2000	Min. : 1.800	Min. : 1.800	Min. : 1.700
1st Qu.:2004	1st Qu.: 3.950	1st Qu.: 3.950	1st Qu.: 3.875
Median :2008	Median : 6.650	Median : 6.900	Median : 6.350
Mean :2007	Mean : 7.952	Mean : 8.207	Mean : 7.712
3rd Qu.:2011	3rd Qu.: 9.725	3rd Qu.:10.350	3rd Qu.: 9.625
Max. :2016	Max. :29.200	Max. :30.700	Max. :27.700
Male.Prevalence	Male.Lower.CI	Male.Upper.CI	Female.Prevalence
Min. :11.50	Min. :12.20	Min. :13.70	Min. :2.700
1st Qu.:13.70	1st Qu.:14.85	1st Qu.:16.07	1st Qu.:3.050
Median :18.40	Median :20.20	Median :21.55	Median :4.000
Mean :18.71	Mean :19.22	Mean :20.62	Mean :4.271
3rd Qu.:23.55	3rd Qu.:22.93	3rd Qu.:24.32	3rd Qu.:5.250
Max. :26.60	Max. :25.80	Max. :27.40	Max. :6.600
NA's :35	NA's :36	NA's :36	NA's :35
Female.Lower.CI	Female.Upper.CI	Non.hispanic.white.Prevalence	
Min. :2.600	Min. :3.300	Min. : 7.70	
1st Qu.:3.100	1st Qu.:3.700	1st Qu.: 9.80	
Median :4.200	Median :4.850	Median :12.20	

[Tips] We notice missing data in a few Prevalence variables.

```
In [85]: # Calculate average Prevalence, no error
```

```
mean(ASD_National$Prevalence)  
mean(ASD_National$Prevalence[ASD_National$Source == 'addm'])  
mean(ASD_National$Prevalence[ASD_National$Source == 'medi'])  
mean(ASD_National$Prevalence[ASD_National$Source == 'nsch'])  
mean(ASD_National$Prevalence[ASD_National$Source == 'sped'])
```

7.95238095238095

10.9875

4.67692307692308

19.025

6.42352941176471

```
In [86]: # Calculate average Male.Prevalence, there is error!
```

```
mean(ASD_National$Male.Prevalence)
```

<NA>

```
In [87]: # Because of NA, mean() cannot process, thus we use na.rm to ignore NAs
```

```
mean(ASD_National$Male.Prevalence, na.rm = TRUE)
```

18.7142857142857

```
In [88]: mean(ASD_National$Female.Prevalence, na.rm = TRUE)
```

4.27142857142857

```
In [89]: # Count occurrences of uniques values in a variable/column: number of rows (of  
table(ASD_National$Year) # ?table
```

```
2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 20  
15  
3 2 3 2 4 2 3 2 4 2 3 2 4 1 2  
1  
2016  
2
```

Data Summarization - Summary of categorical variables

```
In [90]: # List of categorical variables  
names(select_if(ASD_National, is.factor)) # All categorical variables are fact  
names(select_if(ASD_National, is.character)) # No categorical variable is char  
  
'Source' 'Source_Full1' 'Source_Full2' 'Source_UC' 'Source_Full3' 'Prevalence_Risk2'  
'Prevalence_Risk4' 'Year_Factor'
```

```
In [91]: # Look at summary  
summary(select_if(ASD_National, is.factor))
```

```
Source  
addm: 8 Autism & Developmental Disabilities Monitoring Network: 8  
medi:13 Medicaid :13  
nsch: 4 National Survey of Children's Health : 4  
sped:17 Special Education Child Count :17
```

```
Source_Full1  
addm-Autism & Developmental Disabilities Monitoring Network: 8 ADDM: 8  
medi-Medicaid :13 MEDI:13  
nsch-National Survey of Children's Health : 4 NSCH: 4  
sped-Special Education Child Count :17 SPED:17
```

```
Source_Full2 Source_UC  
ADDM Autism & Developmental Disabilities Monitoring Network: 8 ADDM: 8  
MEDI Medicaid :13 MEDI:13  
NSCH National Survey of Children's Health : 4 NSCH: 4  
SPED Special Education Child Count :17 SPED:17
```

```
Prevalence_Risk2 Prevalence_Risk4 Year_Factor  
Low :14 Low :14 2004 : 4  
High:28 Medium :18 2008 : 4  
High : 8 High : 8 2012 : 4  
Very High: 2 Very High: 2 2000 : 3  
2002 : 3  
2006 : 3  
(Other):21
```

```
In [92]: summary(select_if(ASD_National, is.character))
```

< table of extent 0 x 0 >

```
In [93]: # Count occurrences of uniques values in a variable/column  
table(ASD_National$Source)
```

addm	medi	nsch	sped
8	13	4	17

```
In [94]: table(ASD_National$Source_Full3)
```

ADDM Autism & Developmental Disabilities Monitoring Network	
	8
MEDI Medicaid	
	13
NSCH National Survey of Children's Health	
	4
SPED Special Education Child Count	
	17

```
In [95]: table(ASD_National$Year_Factor)
```

2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
3	2	3	2	4	2	3	2	4	2	3	2	4	1	2	
1															
2016															
2															

```
In [96]: table(ASD_National$Prevalence) # numeric is also possible
```

1.8	2.1	2.3	2.6	2.8	3	3.5	3.6	3.9	4.1	4.4	4.8	5.1	5.4	5.6	
5.9															
1	1	1	2	1	2	1	1	1	1	1	1	1	1	1	
1															
6.2	6.4	6.6	6.7	7	7.1	7.7	8	8.2	8.4	9	9.1	9.5	9.8	10.5	1
1.2															
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
1															
11.3	11.9	14.7	14.8	16.2	16.8	21.2	29.2								
1	1	1	1	1	1	1	1								

```
In [97]: # Display unique values (levels) of a factor categorical
lapply(select_if(ASD_National, is.factor), levels)
```

\$Source

'addm' 'medi' 'nsch' 'sped'

\$Source_Full1

'Autism & Developmental Disabilities Monitoring Network' 'Medicaid'
'National Survey of Children's Health' 'Special Education Child Count'

\$Source_Full2

'addm-Autism & Developmental Disabilities Monitoring Network' 'medi-Medicaid'
'nsch-National Survey of Children's Health' 'sped-Special Education Child Count'

\$Source_UC

'ADDM' 'MEDI' 'NSCH' 'SPED'

\$Source_Full3

'ADDM Autism & Developmental Disabilities Monitoring Network' 'MEDI Medicaid'
'NSCH National Survey of Children's Health' 'SPED Special Education Child Count'

\$Prevalence_Risk2

'Low' 'High'

\$Prevalence_Risk4

'Low' 'Medium' 'High' 'Very High'

\$Year_Factor

'2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009' '2010' '2011' '2012'
'2013' '2014' '2015' '2016'

```
In [98]: # or using variable names
```

```
lapply(ASD_National[c('Source_UC', 'Year_Factor')], levels)
```

\$Source_UC

'ADDM' 'MEDI' 'NSCH' 'SPED'

\$Year_Factor

'2000' '2001' '2002' '2003' '2004' '2005' '2006' '2007' '2008' '2009' '2010' '2011' '2012'
'2013' '2014' '2015' '2016'

```
In [99]: # Pivot of counting occurrences
table(ASD_National$Source_Full3, ASD_National$Year) # table(ASD_National$Year,
```

		2000	2001	200
2	ADDM Autism & Developmental Disabilities Monitoring Network	1	0	
1	MEDI Medicaid	1	1	
1	NSCH National Survey of Children's Health	0	0	
0	SPED Special Education Child Count	1	1	
1				
		2003	2004	200
5	ADDM Autism & Developmental Disabilities Monitoring Network	0	1	
0	MEDI Medicaid	1	1	
1	NSCH National Survey of Children's Health	0	1	
~				

```
In [100]: # Pivot of counting occurrences
table(ASD_National$Prevalence_Risk2, ASD_National$Source)

# Pivot of counting occurrences
table(ASD_National$Prevalence_Risk4, ASD_National$Source)
```

	addm	medi	nsch	sped
Low	0	7	0	7
High	8	6	4	10

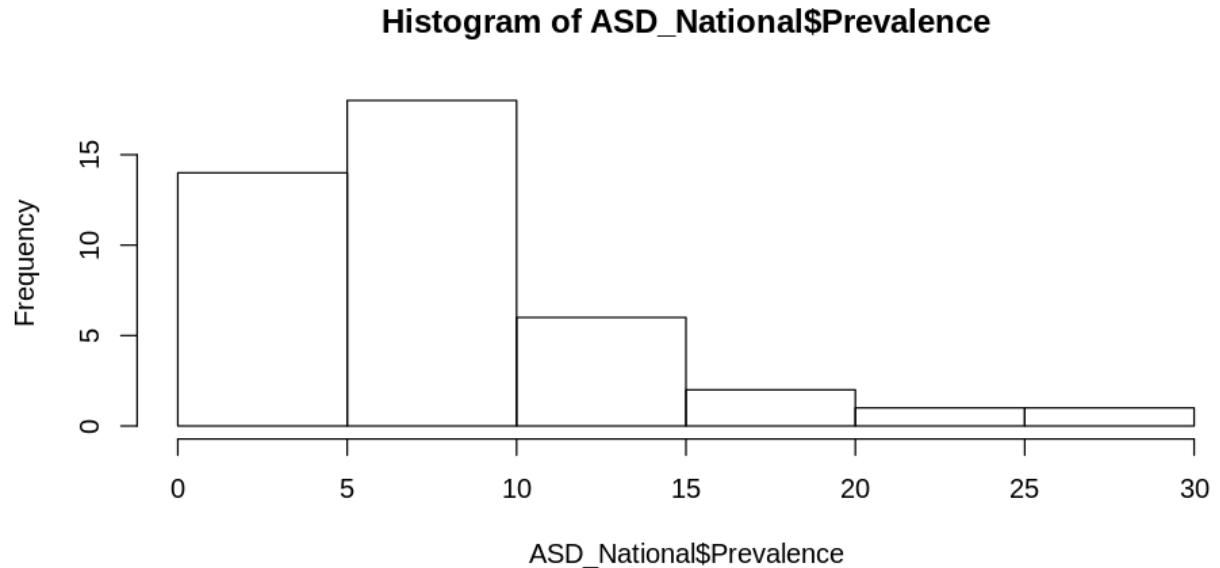
	addm	medi	nsch	sped
Low	0	7	0	7
Medium	4	6	1	7
High	4	0	1	3
Very High	0	0	2	0

Data Visualisation (Base Graphic)

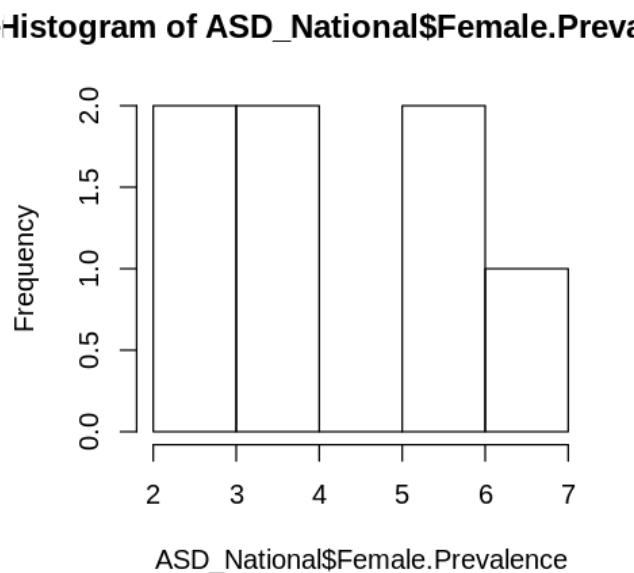
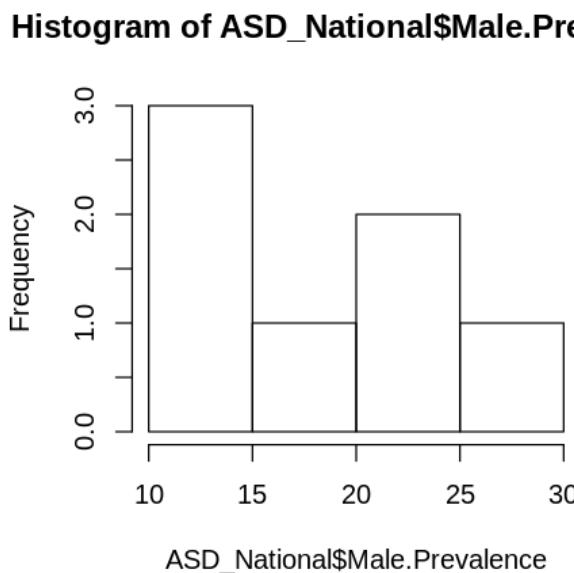
```
In [101]: # library(repr)
# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)
```

Data Visualisation (Base Graphic) - Histogram (distribution of binned continuous variable)

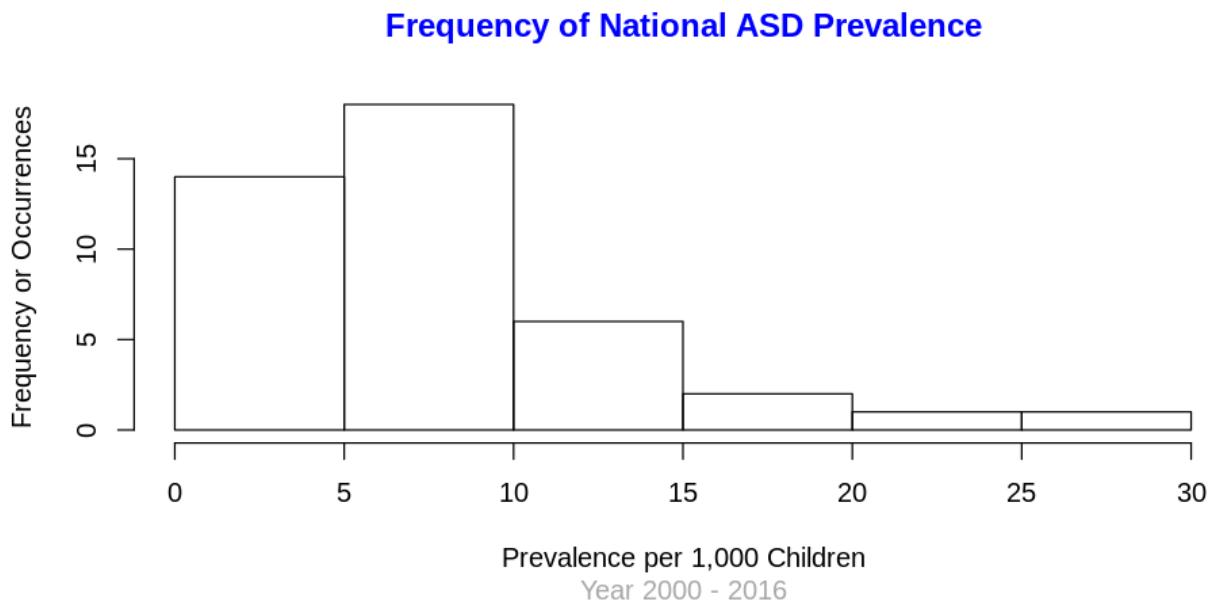
```
In [102]: hist(ASD_National$Prevalence)
```



```
In [103]: par(mfrow=c(1, 2)) # multiple plots on one page: row split to: 1, column split  
hist(ASD_National$Male.Prevalence)  
hist(ASD_National$Female.Prevalence)  
par(mfrow=c(1, 1)) # Reset to one plot on one page
```



```
In [104]: # Histogram with annotations  
hist(ASD_National$Prevalence,  
      main = "Frequency of National ASD Prevalence", # Chart title  
      xlab = "Prevalence per 1,000 Children", # x axis label  
      ylab = "Frequency or Occurrences",# y axis label  
      sub = "Year 2000 - 2016", # Chart subtitle at bottom  
      col.main="blue", col.lab="black", col.sub="darkgrey") # Colours
```



Density plot (distribution for continuous variable normalized to 100% area under curve)

<https://www.statmethods.net/graphs/density.html> (<https://www.statmethods.net/graphs/density.html>)

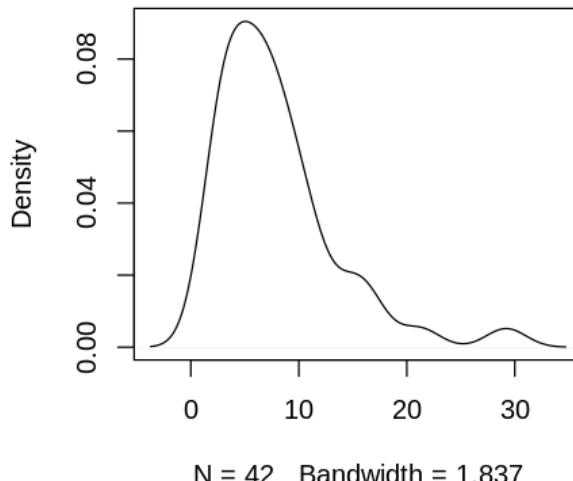
```
In [105]: par(mfrow=c(1, 2)) # multiple plots on one page: row split to: 1, column split

plot(density(ASD_National$Prevalence))

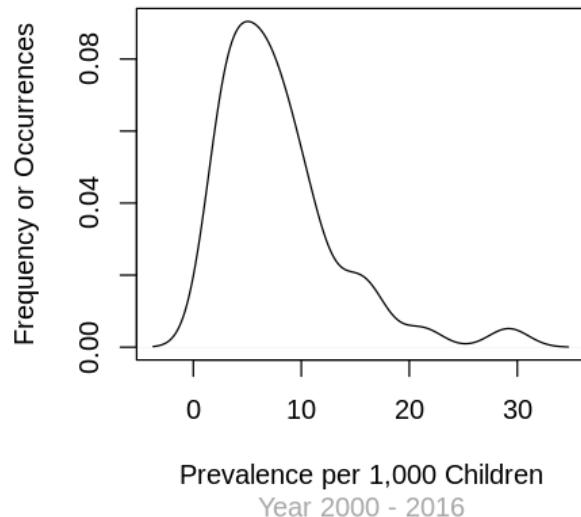
# Density plot with annotations
plot(density(ASD_National$Prevalence),
     main = "Density of National ASD Prevalence",
     xlab = "Prevalence per 1,000 Children",
     ylab = "Frequency or Occurrences",
     sub = "Year 2000 - 2016",
     col.main="blue", col.lab="black", col.sub="darkgrey")

par(mfrow=c(1, 1))
```

density.default(x = ASD_National\$Prevalence)



Density of National ASD Prevalence



Boxplot plot (median, 25% quantile, 75% quantile)

<https://www.statmethods.net/graphs/boxplot.html> (<https://www.statmethods.net/graphs/boxplot.html>)

<https://stats.stackexchange.com/questions/156778/percentile-vs-quantile-vs-quartile>
(<https://stats.stackexchange.com/questions/156778/percentile-vs-quantile-vs-quartile>)

0 quartile = 0 quantile = 0 percentile

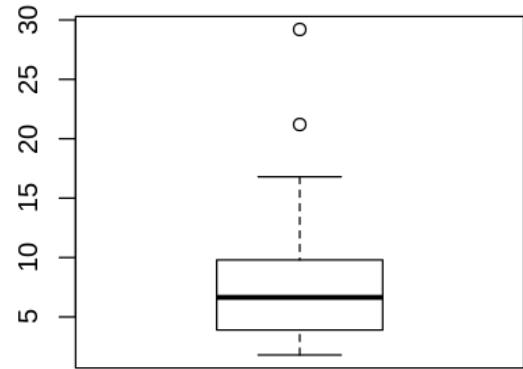
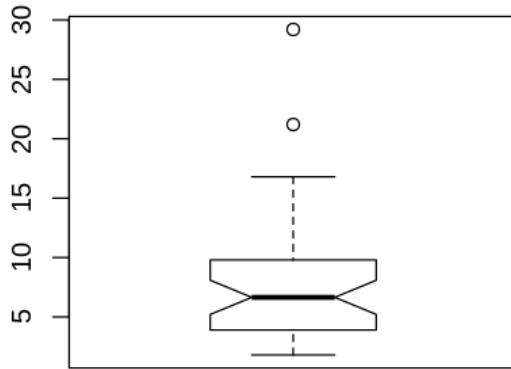
1 quartile = 0.25 quantile = 25 percentile

2 quartile = .5 quantile = 50 percentile (median)

3 quartile = .75 quantile = 75 percentile

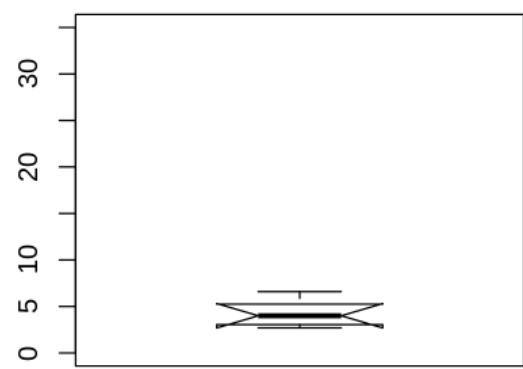
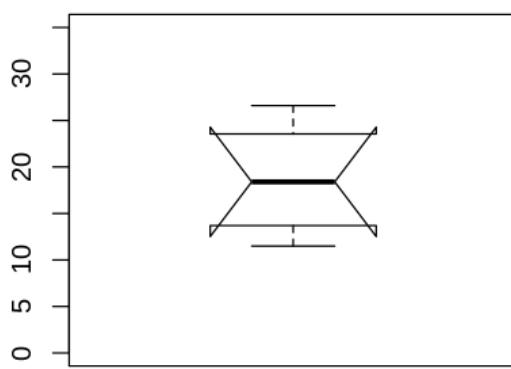
4 quartile = 1 quantile = 100 percentile

```
In [106]: par(mfrow=c(1, 2)) # multiple plots on one page: row split to: 1, column split
# All children prevalence with and without 95% confidence side by side:
boxplot(ASD_National$Prevalence, notch = TRUE) # 95% confidence interval - a n
boxplot(ASD_National$Prevalence) # All children
par(mfrow=c(1, 1))
```



```
In [107]: par(mfrow=c(1, 2)) # multiple plots on one page: row split to: 1, column split
# Male prevalence and Female prevalence side by side:
boxplot(ASD_National$Male.Prevalence, ylim = c(0, 35), notch = TRUE) # Male ch
boxplot(ASD_National$Female.Prevalence, ylim = c(0, 35), notch = TRUE) # Femal
par(mfrow=c(1, 1))
```

Warning message in bxp(list(stats = structure(c(11.5, 13.7, 18.4, 23.55, 26.6), .Dim = c(5L, :
"some notches went outside hinges ('box'): maybe set notch=FALSE")
Warning message in bxp(list(stats = structure(c(2.7, 3.05, 4, 5.25, 6.6), .Dim = c(5L, :
"some notches went outside hinges ('box'): maybe set notch=FALSE")



```
In [108]: # Display value ranges  
# numeric:  
range(ASD_National$Prevalence)
```

1.8 29.2

```
In [109]: range(ASD_National$Year)
```

2000 2016

```
In [110]: # categorical:  
min(ASD_National$Year_Factor)
```

2000

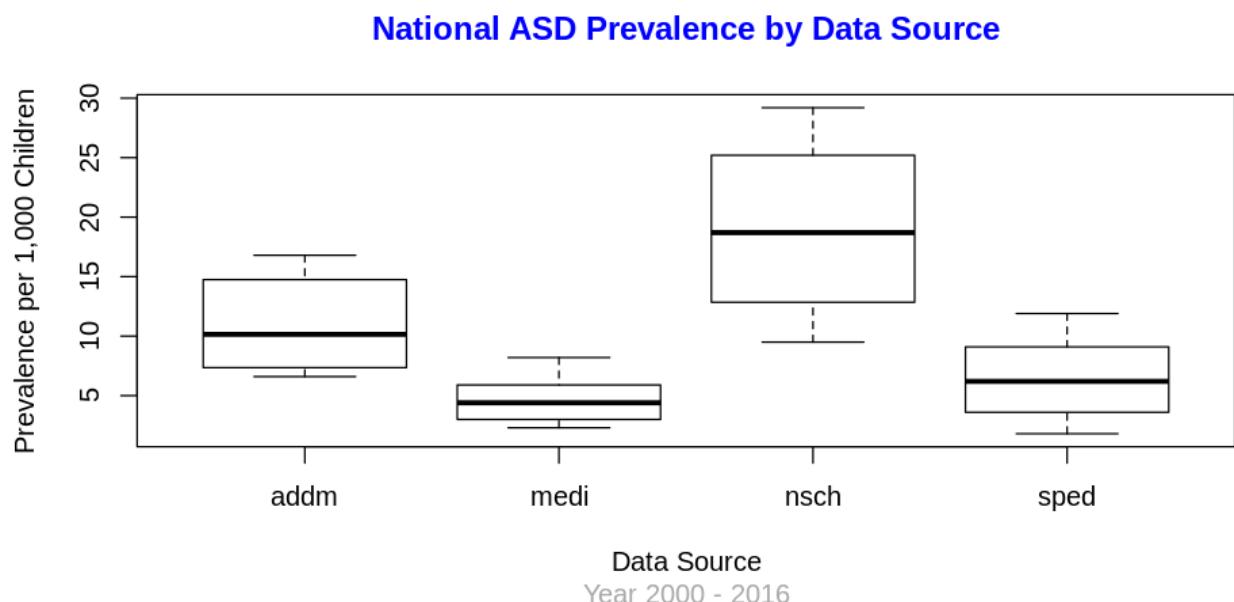
► Levels:

```
In [111]: max(ASD_National$Year_Factor)
```

2016

► Levels:

```
In [112]: # Create 'Prevalence' box plots break by 'Source'  
boxplot(ASD_National$Prevalence ~ ASD_National$Source,  
        main = "National ASD Prevalence by Data Source",  
        xlab = "Data Source",  
        ylab = "Prevalence per 1,000 Children",  
        sub = "Year 2000 - 2016",  
        col.main="blue", col.lab="black", col.sub="darkgrey")
```



Quiz:

Set `notch=TRUE` to above boxplot. Are there overlapping among four data sources?

```
In [113]: # Write your code below and press Shift+Enter to execute
```

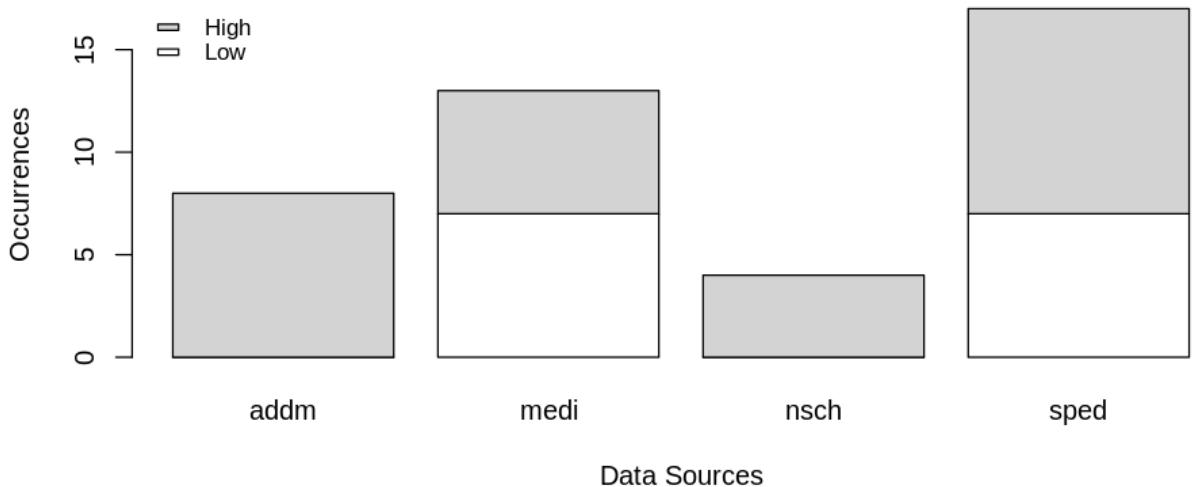
Double-click **here** for the solution.

Data Visualisation (Base Graphic) - Bar plot

```
In [114]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)

In [115]: # -----
# [National] Risk by Data Source
# -----
# Create bar chart using R graphics
counts = table(ASD_National$Prevalence_Risk2, ASD_National$Source)
#counts = table(ASD_National$Source, ASD_National$Prevalence_Risk4)
barplot(counts,
        main="Prevalence by Data Sources and Risk Levels",
        xlab="Data Sources", col=c("white", "lightgrey"),
        ylab="Occurrences",
        legend = rownames(counts),
        args.legend = list(x="topleft", bty = "n", cex = 0.85, y.intersp=2))
```

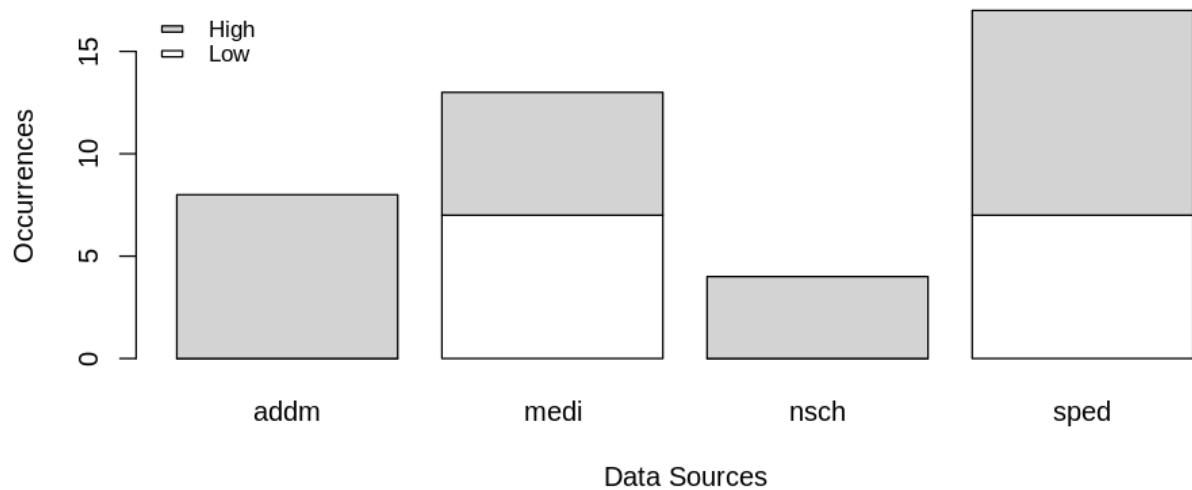
Prevalence by Data Sources and Risk Levels



In [116]:

```
# -----  
# [National] Risk by Data Source  
# -----  
# Create bar chart using R graphics  
counts = table(ASD_National$Prevalence_Risk2, ASD_National$Source) # Count of  
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 = 2))
```

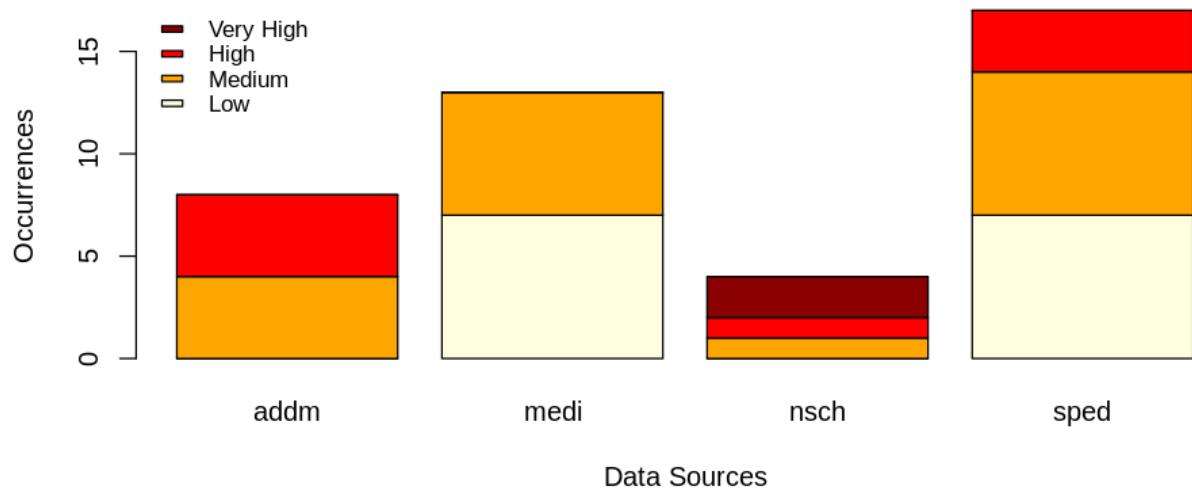
Prevalence by Data Sources and Risk Levels



In [117]:

```
# -----  
# [National] Risk by Data Source  
# -----  
# Create bar chart using R graphics  
counts = table(ASD_National$Prevalence_Risk4, ASD_National$Source) # Count of  
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 = 2))
```

Prevalence Occurrence by Source and Risk



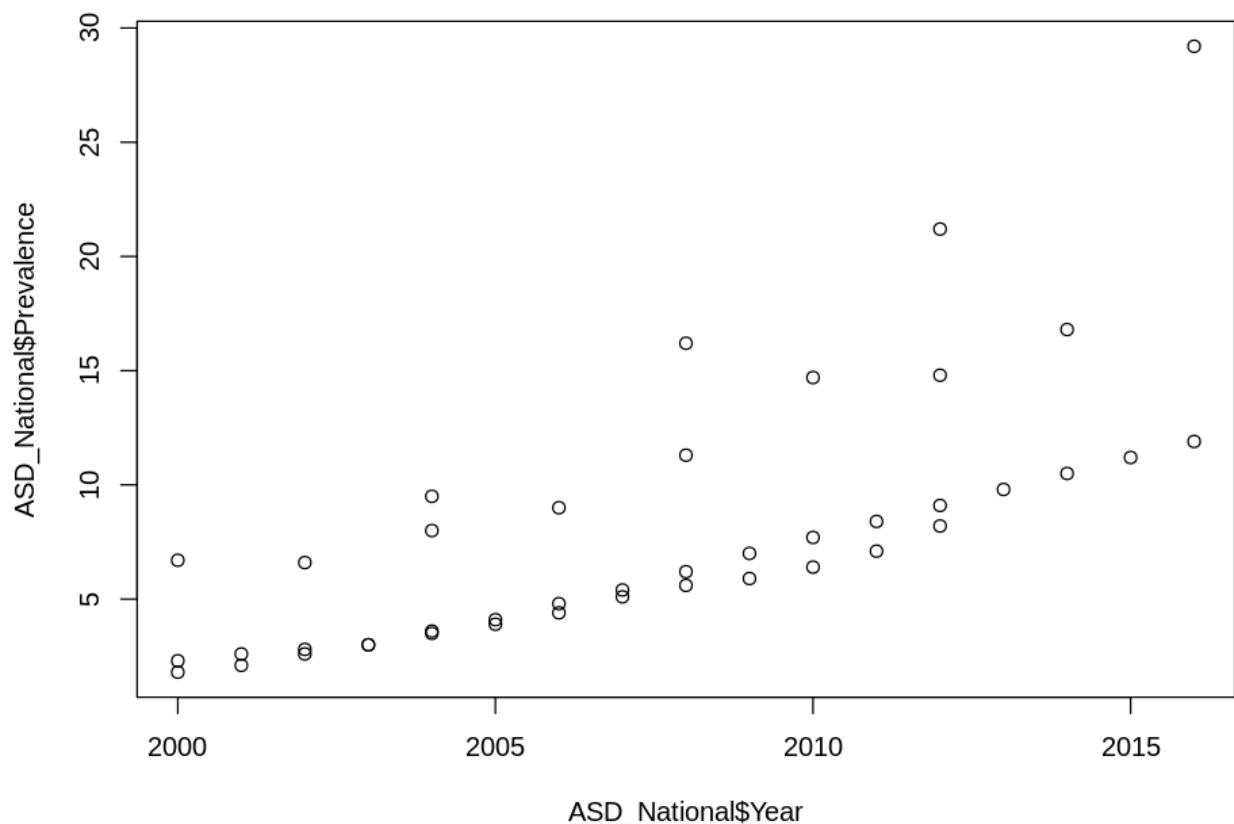
Data Visualisation (Base Graphic) - Line chart

In [118]:

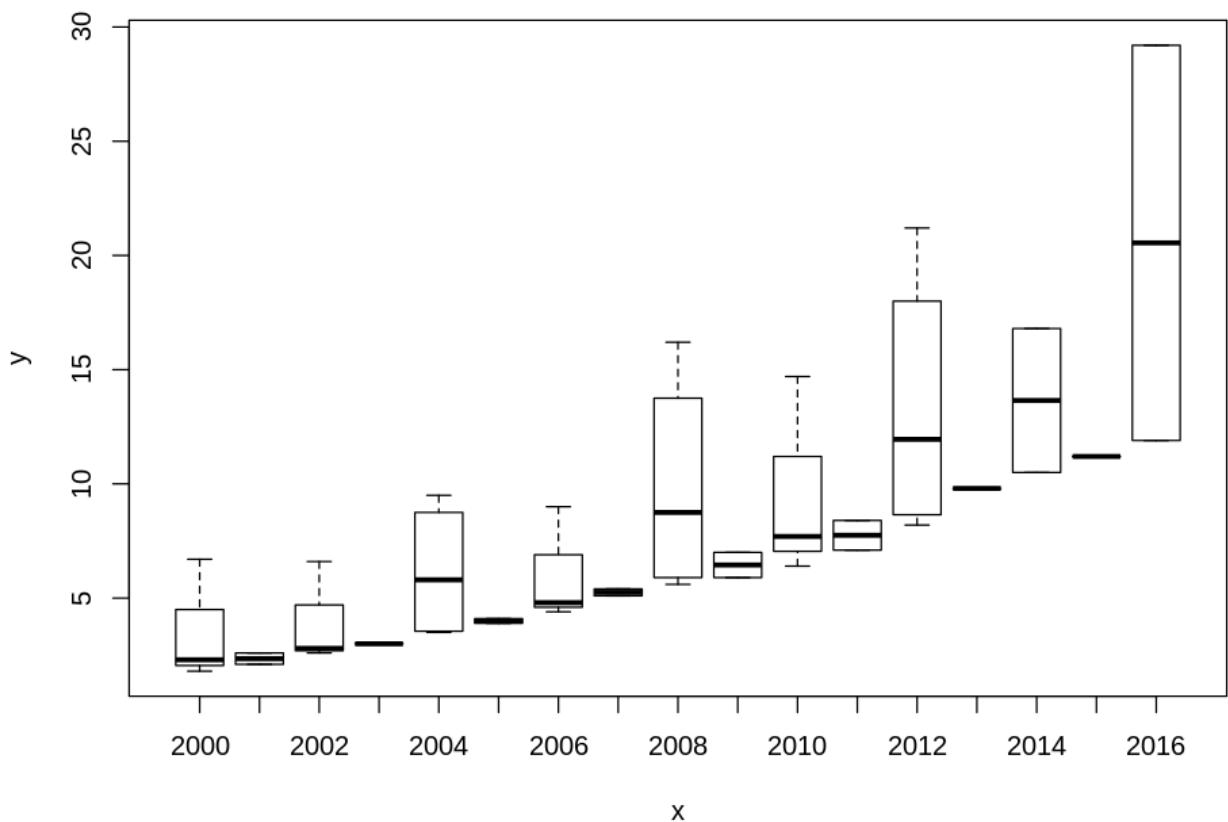
```
# Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=6)
```

In [119]:

```
# -----  
# [National] < Prevalence has changed over Time >  
# -----  
# Prevalence over Year  
# Use Year      as x-axis: y value Prevalence is NOT aggregated for different countries  
plot(ASD_National$Year, ASD_National$Prevalence)
```



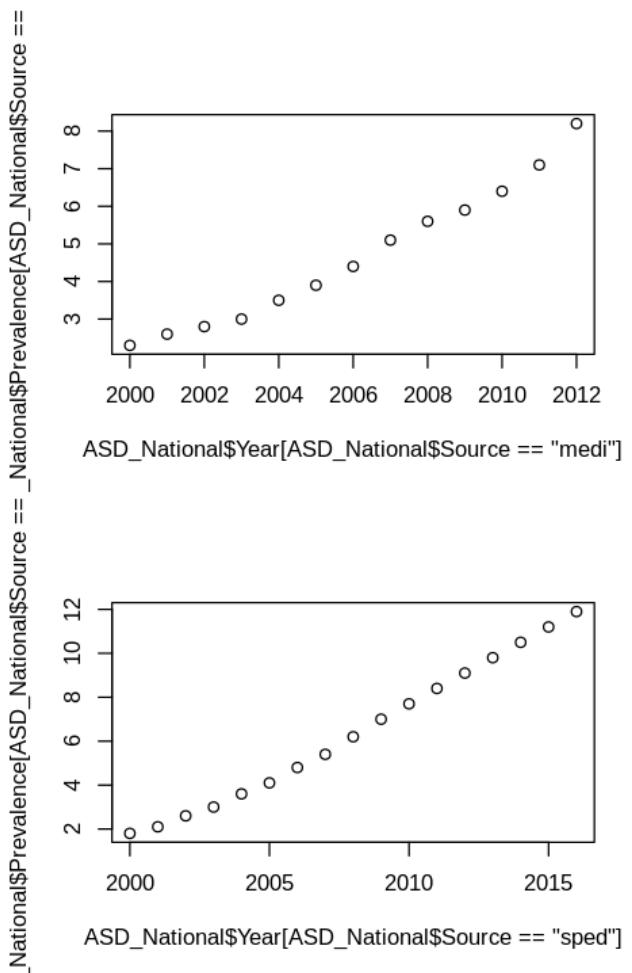
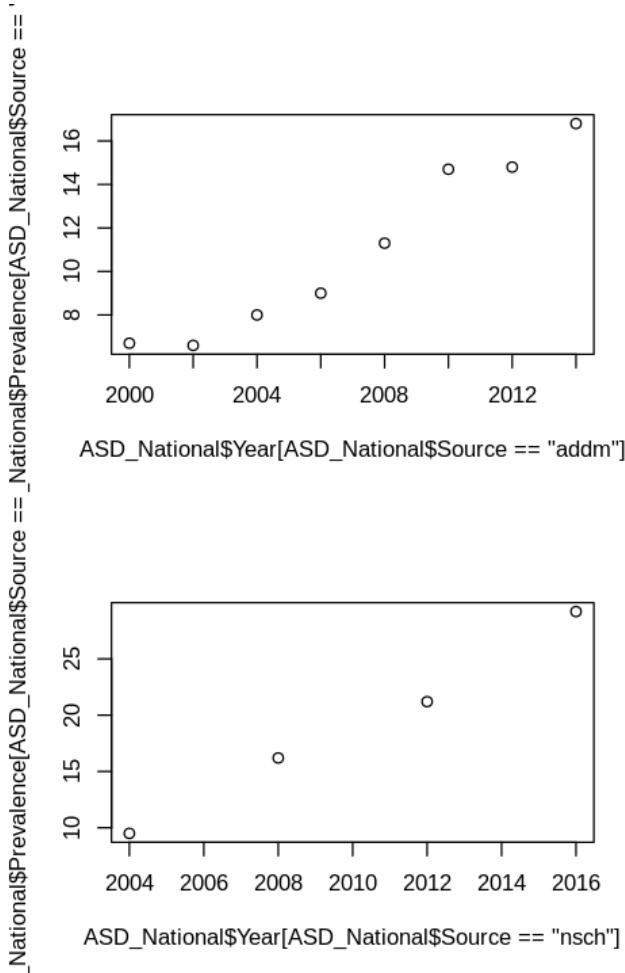
```
In [120]: # Use Year_factor as x-axis: y value Prevalence is aggregated for different plot(ASD_National$Year_Factor, ASD_National$Prevalence)
```



```
In [121]: # table(ASD_National$Source_Full3)
```

```
In [122]: par(mfrow=c(2, 2))
```

```
# Prevalence over Year, from data source:  
# addm-Autism & Developmental Disabilities Monitoring Network  
plot(ASD_National$Year[ASD_National$Source == 'addm'],  
     ASD_National$Prevalence[ASD_National$Source == 'addm'])  
  
# Prevalence over Year, from data source:  
# medi-Medicaid  
plot(ASD_National$Year[ASD_National$Source == 'medi'],  
     ASD_National$Prevalence[ASD_National$Source == 'medi'])  
  
# Prevalence over Year, from data source:  
# nsch-National Survey of Children Health  
plot(ASD_National$Year[ASD_National$Source == 'nsch'],  
     ASD_National$Prevalence[ASD_National$Source == 'nsch'])  
  
# Prevalence over Year, from data source:  
# sped-Special Education Child Count  
plot(ASD_National$Year[ASD_National$Source == 'sped'],  
     ASD_National$Prevalence[ASD_National$Source == 'sped'])  
  
par(mfrow=c(1, 1)) # Reset to one plot on one page
```



In [123]:

```
# -----
# Add more annotations to above plots
# -----
# Color list
# addm : darkblue
# medi : orange
# nsch : darkred
# sped : skyblue

par(mfrow=c(2, 2))

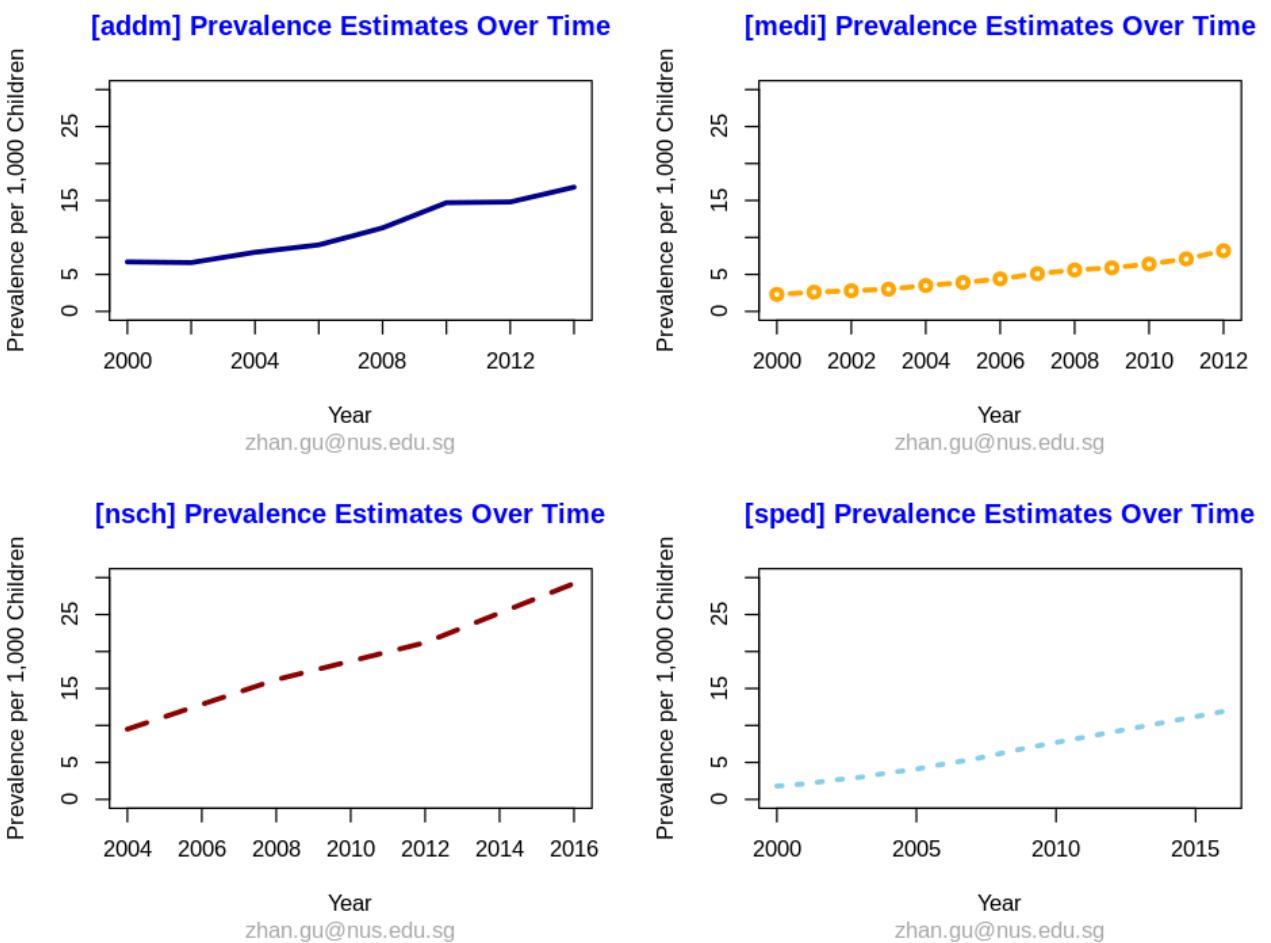
# Prevalence over Year, from data source:
# addm-Autism & Developmental Disabilities Monitoring Network
plot(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Prevalence[ASD_National$Source == 'addm'],
      type="l", # dot/point type
      lty=1, # line type
      lwd=3, # line width
      col="darkblue", # line color
      xlab="Year",
      ylab="Prevalence per 1,000 Children",
      ylim = c(0, 30), # Set value range of y axis
      main ="[addm] Prevalence Estimates Over Time",
      sub  = "zhan.gu@nus.edu.sg",
      col.main="blue", col.lab="black", col.sub="darkgrey")

# Prevalence over Year, from data source:
# medi-Medicaid
plot(ASD_National$Year[ASD_National$Source == 'medi'],
      ASD_National$Prevalence[ASD_National$Source == 'medi'],
      type="b", lty=1, lwd=3, col="orange",
      xlab="Year",
      ylab="Prevalence per 1,000 Children",
      ylim = c(0, 30), # Set value range of y axis
      main ="[medi] Prevalence Estimates Over Time",
      sub  = "zhan.gu@nus.edu.sg",
      col.main="blue", col.lab="black", col.sub="darkgrey")

# Prevalence over Year, from data source:
# nsch-National Survey of Children Health
plot(ASD_National$Year[ASD_National$Source == 'nsch'],
      ASD_National$Prevalence[ASD_National$Source == 'nsch'],
      type="l", lty=2, lwd=3, col="darkred",
      xlab="Year",
      ylab="Prevalence per 1,000 Children",
      ylim = c(0, 30), # Set value range of y axis
      main ="[nsch] Prevalence Estimates Over Time",
      sub  = "zhan.gu@nus.edu.sg",
      col.main="blue", col.lab="black", col.sub="darkgrey")

# Prevalence over Year, from data source:
# sped-Special Education Child Count
plot(ASD_National$Year[ASD_National$Source == 'sped'],
      ASD_National$Prevalence[ASD_National$Source == 'sped'],
      type="l", lty=3, lwd=3, col="skyblue",
      xlab="Year",
      ylab="Prevalence per 1,000 Children",
      ylim = c(0, 30), # Set value range of y axis
      main ="[sped] Prevalence Estimates Over Time",
      sub  = "zhan.gu@nus.edu.sg",
      col.main="blue", col.lab="black", col.sub="darkgrey")

par(mfrow=c(1, 1)) # Reset to one plot on one page
```



Data Visualisation (Base Graphic) - [R] REPORTED PREVALENCE HAS CHANGED OVER TIME by [Data Source]

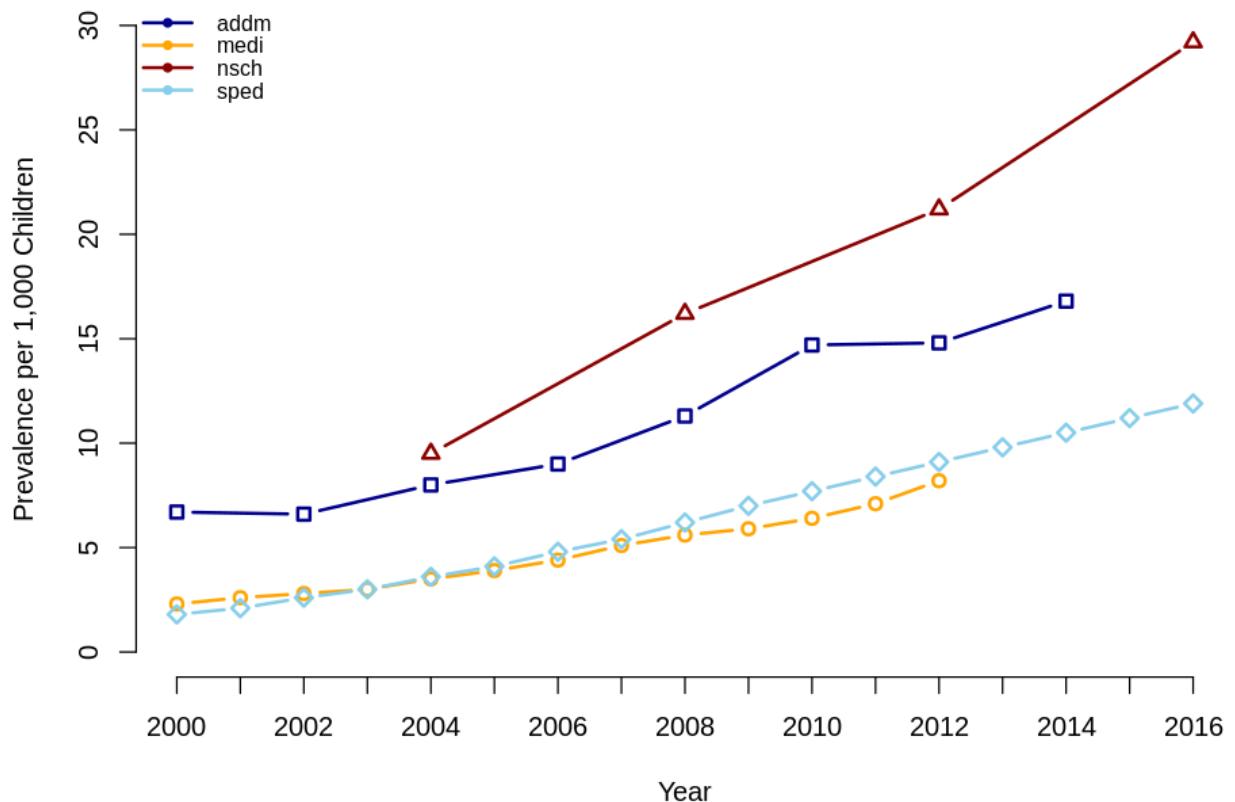
Create multiple lines within a single chart

In [124]:

```
# -----
# [National] < Prevalence Varies over Time/Year by Data Source >
# -----
# Create a first line
plot(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Prevalence[ASD_National$Source == 'addm'],
      col = "darkblue", lty = 1, lwd = 2,
      type = "b", # use dot/point
      pch = 0, # dot/point type: http://www.endmemo.com/program/R/pchsymbols.php
      xlab="Year",
      xlim=c(2000, 2016), # Set x axis value range
      ylab="Prevalence per 1,000 Children",
      ylim=c(0, 30), # Set y axis value range
      main="Prevalence Estimates Over Time by Data Source",
      col.main="black", col.lab="black", col.sub="grey",
      frame = FALSE, # Remove frame
      axes=FALSE # Remove x and y axis
)
axis(1, at=seq(2000, 2016, 1)) # Customize x axis
axis(2, at=seq(0, 30, 5)) # Customize y axis

# Add another line
lines(ASD_National$Year[ASD_National$Source == 'medi'],
      ASD_National$Prevalence[ASD_National$Source == 'medi'],
      pch = 1, col = "orange", type = "b", lty = 1, lwd = 2
)
# Add another line
lines(ASD_National$Year[ASD_National$Source == 'nsch'],
      ASD_National$Prevalence[ASD_National$Source == 'nsch'],
      pch = 2, col = "darkred", type = "b", lty = 1, lwd = 2
)
# Add another line
lines(ASD_National$Year[ASD_National$Source == 'sped'],
      ASD_National$Prevalence[ASD_National$Source == 'sped'],
      pch = 5, col = "skyblue", type = "b", lty = 1, lwd = 2
)
# Add a legend to the plot
legend("topleft", legend=levels(ASD_National$Source),
       col=c("darkblue", "orange", "darkred", "skyblue"),
       pch = 20, # dot in a line
       lty = 1, # line type
       lwd = 2, # line width
       cex=0.8, # size of text
       bty = 'n' # Without frame
)
```

Prevalence Estimates Over Time by Data Source



R pch: dot/point type: <http://www.endmemo.com/program/R/pchsymbols.php>
(<http://www.endmemo.com/program/R/pchsymbols.php>).

R plot colour list: <https://www.r-graph-gallery.com/42-colors-names.html> (<https://www.r-graph-gallery.com/42-colors-names.html>).

Data Visualisation (Base Graphic) - [R] REPORTED PREVALENCE VARIES BY SEX [Source: ADDM] over [Year]

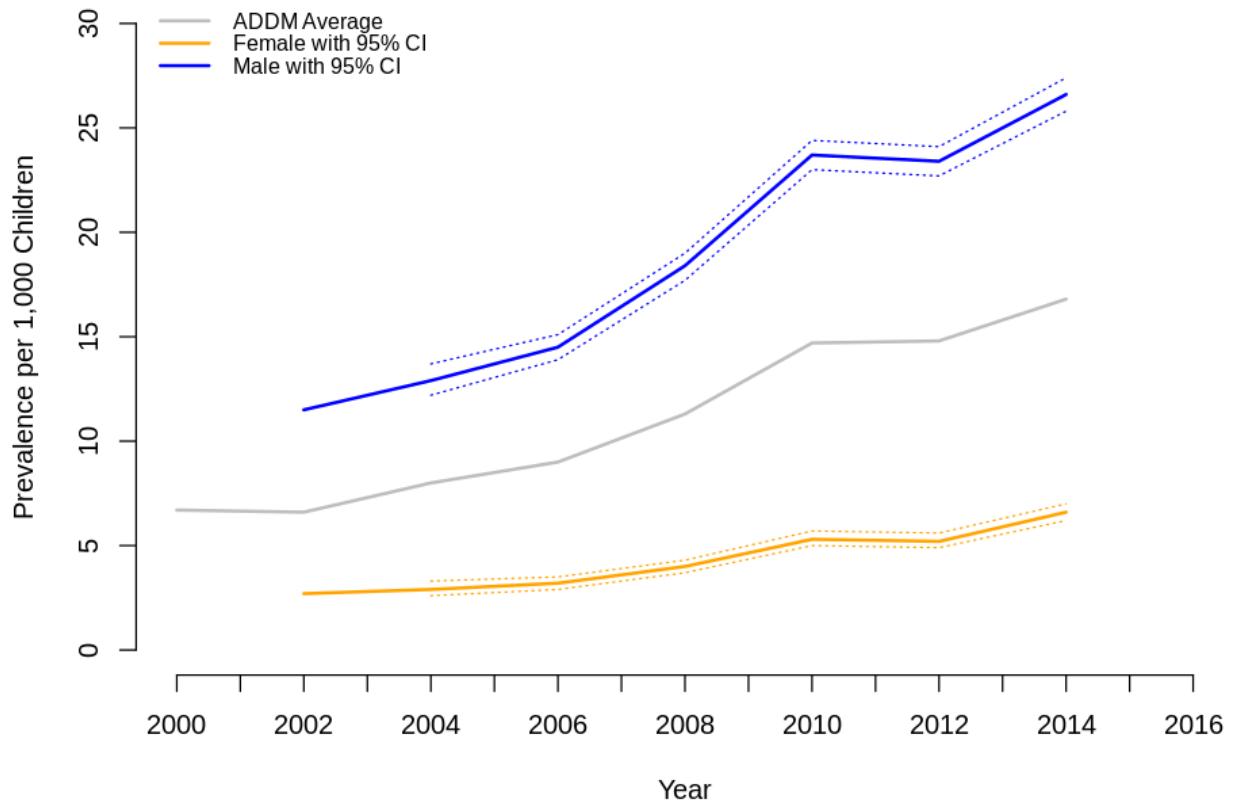
In [125]:

```
# -----
# [addm] < Prevalence Varies by Sex >
# -----
# Create a first line
plot(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Prevalence[ASD_National$Source == 'addm'],
      col = "grey", lty = 1, lwd = 2,
      type = "l", # use dot/point
      pch = 0, # dot/point type: http://www.endmemo.com/program/R/pchsymbols.php
      xlab="Year",
      xlim=c(2000, 2016), # Set x axis value range
      ylab="Prevalence per 1,000 Children",
      ylim=c(0, 30), # Set y axis value range
      main="Prevalence Estimates by Sex [ADDM]",
      col.main="black", col.lab="black", col.sub="grey",
      frame = FALSE, # Remove frame
      axes=FALSE # Remove x and y axis
)
axis(1, at=seq(2000, 2016, 1)) # Customize x axis
axis(2, at=seq(0, 30, 5)) # Customize y axis

# Add Female prevalence
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Female.Prevalence[ASD_National$Source == 'addm'],
      pch = 1, col = "orange", type = "l", lty = 1, lwd = 2)
# Add Female prevalence lower CI
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Female.Lower.CI[ASD_National$Source == 'addm'],
      pch = 1, col = "orange", type = "l", lty = 3, lwd = 1)
# Add Female prevalence upper CI
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Female.Upper.CI[ASD_National$Source == 'addm'],
      pch = 1, col = "orange", type = "l", lty = 3, lwd = 1)

# Add Male prevalence
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Male.Prevalence[ASD_National$Source == 'addm'],
      pch = 1, col = "blue", type = "l", lty = 1, lwd = 2)
# Add Male prevalence lower CI
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Male.Lower.CI[ASD_National$Source == 'addm'],
      pch = 1, col = "blue", type = "l", lty = 3, lwd = 1)
# Add Male prevalence upper CI
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Male.Upper.CI[ASD_National$Source == 'addm'],
      pch = 1, col = "blue", type = "l", lty = 3, lwd = 1)
# Add a legend to the plot
legend("topleft", legend=c('ADDM Average', 'Female with 95% CI', 'Male with 95% CI'),
       col=c("grey", "orange", "blue"),
       #           pch = 20, # dot in a line
       lty = 1, # line type
       lwd = 2, # line width
       cex=0.8, # size of text
       bty = 'n' # Without frame
)
```

Prevalence Estimates by Sex [ADDM]



Data Visualisation (Base Graphic) - [R] REPORTED PREVALENCE VARIES BY RACE AND ETHNICITY [Source: ADDM]

In [126]:

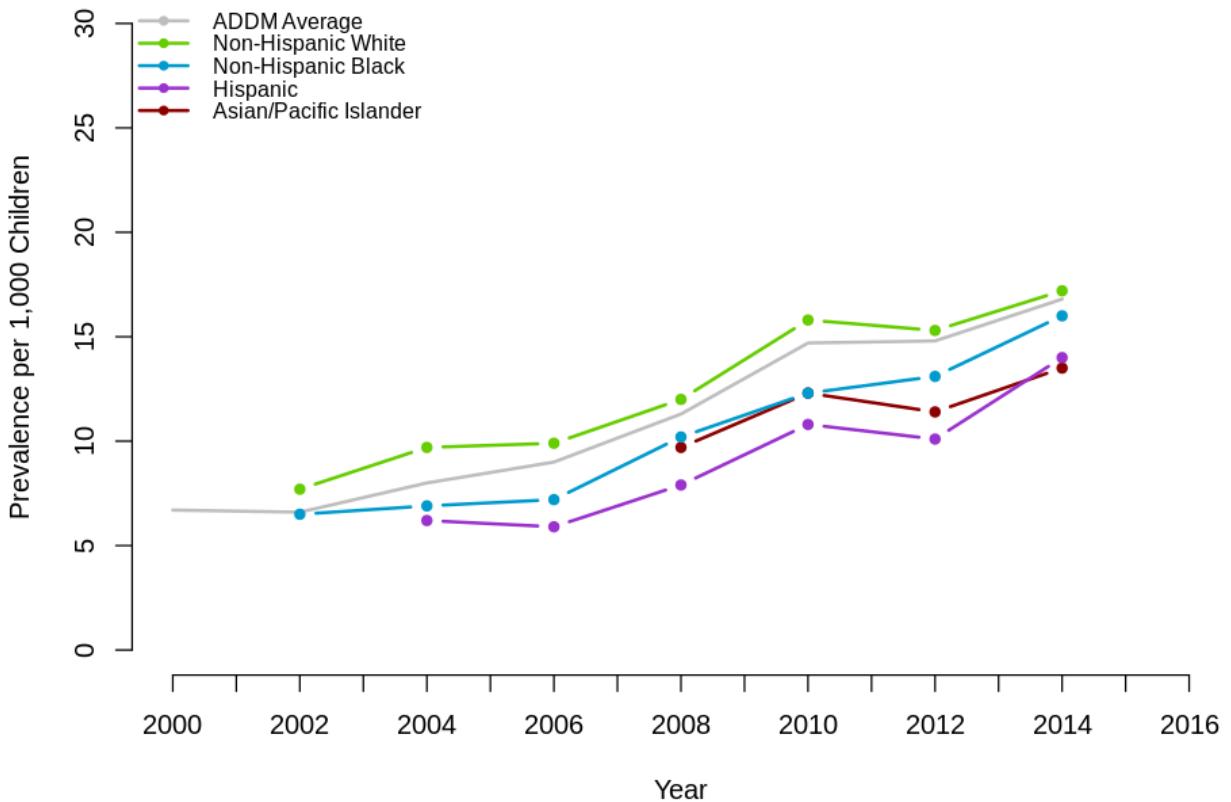
```
# -----
# [addm] < Prevalence Varies by Race and Ethnicity >
# -----
# Create a first line
plot(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Prevalence[ASD_National$Source == 'addm'],
      col = "grey", lty = 1, lwd = 2,
      type = "l", # use dot/point
      pch = 0, # dot/point type: http://www.endmemo.com/program/R/pchsymbols.php
      xlab="Year",
      xlim=c(2000, 2016), # Set x axis value range
      ylab="Prevalence per 1,000 Children",
      ylim=c(0, 30), # Set y axis value range
      main="Prevalence Estimates by Race/Ethnicity [ADDM]",
      col.main="black", col.lab="black", col.sub="grey",
      frame = FALSE, # Remove frame
      axes=FALSE # Remove x and y axis
)
axis(1, at=seq(2000, 2016, 1)) # Customize x axis
axis(2, at=seq(0, 30, 5)) # Customize y axis

# R plot colour list: https://www.r-graph-gallery.com/42-colors-names.html

# Add Asian.or.Pacific.Islander.Prevalence
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Asian.or.Pacific.Islander.Prevalence[ASD_National$Source == 'addm'],
      pch = 20, col = "darkred", type = "b", lty = 1, lwd = 2)
# Add Hispanic.Prevalence
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Hispanic.Prevalence[ASD_National$Source == 'addm'],
      pch = 20, col = "darkorchid3", type = "b", lty = 1, lwd = 2)
# Add Non.hispanic.black.Prevalence
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Non.hispanic.black.Prevalence[ASD_National$Source == 'addm'],
      pch = 20, col = "deepskyblue3", type = "b", lty = 1, lwd = 2)
# Add Non.hispanic.white.Prevalence
lines(ASD_National$Year[ASD_National$Source == 'addm'],
      ASD_National$Non.hispanic.white.Prevalence[ASD_National$Source == 'addm'],
      pch = 20, col = "chartreuse3", type = "b", lty = 1, lwd = 2)

# Add a legend to the plot
legend("topleft", legend=c('ADDM Average',
                           'Non-Hispanic White',
                           'Non-Hispanic Black',
                           'Hispanic',
                           'Asian/Pacific Islander'),
       col=c("grey", "chartreuse3", "deepskyblue3", "darkorchid3", "darkred"),
       pch = 20, # dot in a line
       lty = 1, # line type
       lwd = 2, # line width
       cex=0.8, # size of text
       bty = 'n' # Without frame
)
```

Prevalence Estimates by Race/Ethnicity [ADDM]



```
In [127]: # Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=4)
```

Quiz:

Add 95% Confidence Interval to above plot

```
In [128]: # Write your code below and press Shift+Enter to execute
```

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

Quiz:

Use `table()` to count No. prevalence records for each Data Source. Then use `barplot()` to visualize.

```
In [129]: # Write your code below and press Shift+Enter to execute
```

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

```
In [ ]:
```

Quiz:

Which Data Sources are available in which years?

In [130]: # Write your code below and press Shift+Enter to execute

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

Quiz:

Which Data Source has breakdown Prevalence data by sex/gender?

In [131]: # Write your code below and press Shift+Enter to execute

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

Quiz:

Which Data Source has breakdown Prevalence data by race and ethnicity?

In [132]: # Write your code below and press Shift+Enter to execute

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

Data Visualisation (Enhanced)

In [133]: `if(!require(ggplot2)){install.packages("ggplot2")}
library(ggplot2)`

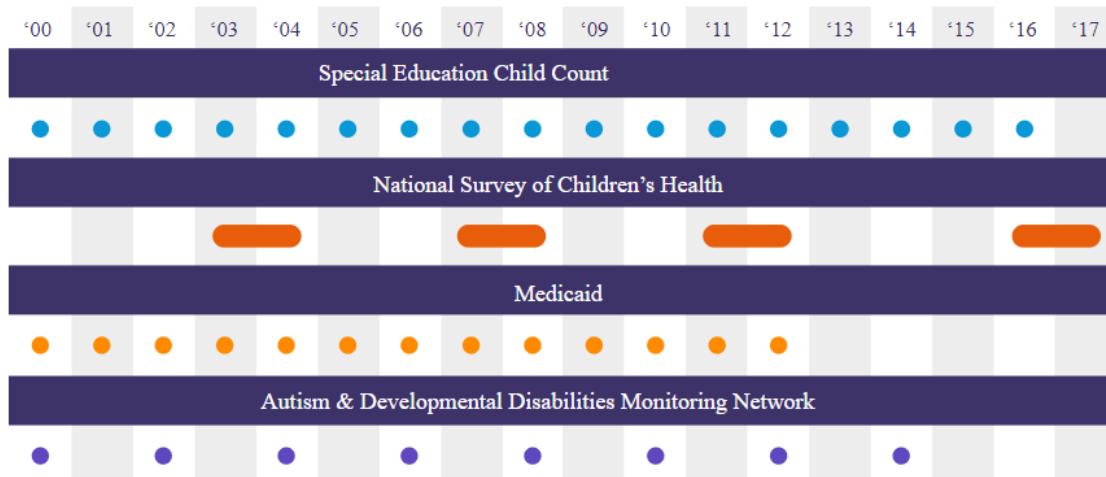
Loading required package: ggplot2

In [134]: `# Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)`

Data Visualisation (Enhanced) - [CDC] Explore the Data

Years Data Available

Select state: ▾



WHY THIS MATTERS

Because ASD data are collected at specific times, they provide a snapshot of what was going on at a certain moment in time. Findings from different data sources are typically reported a year or more *after* the data were collected; therefore, prevalence may have changed between the time data were collected and the time they were reported.

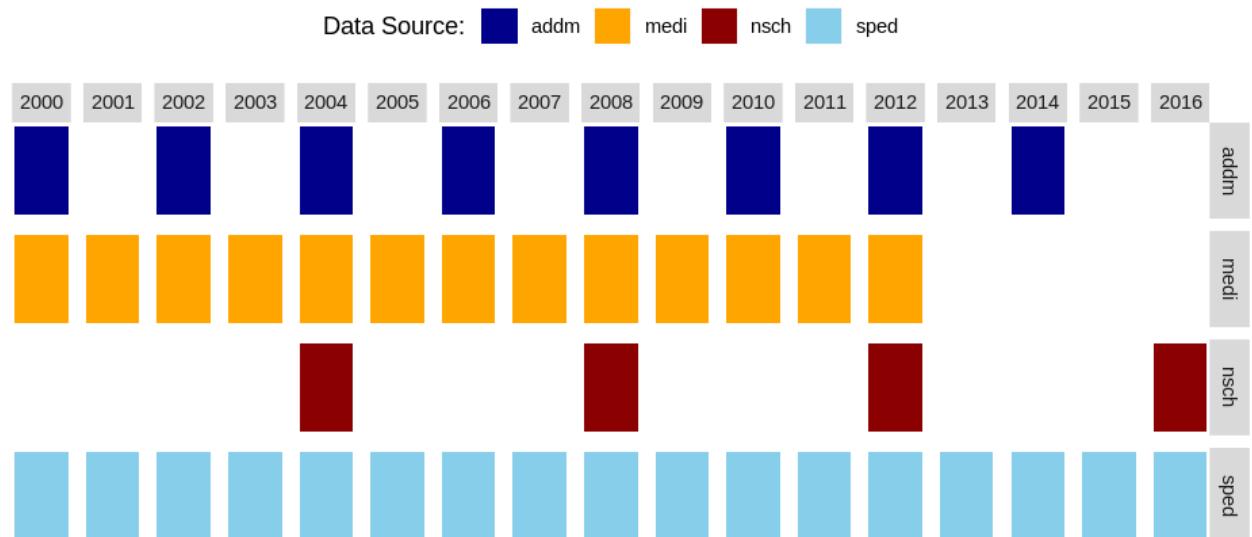
*ADDM estimate = the total for all sites combined.

Data Visualisation (Enhanced) - [R] Explore the Data

In [135]:

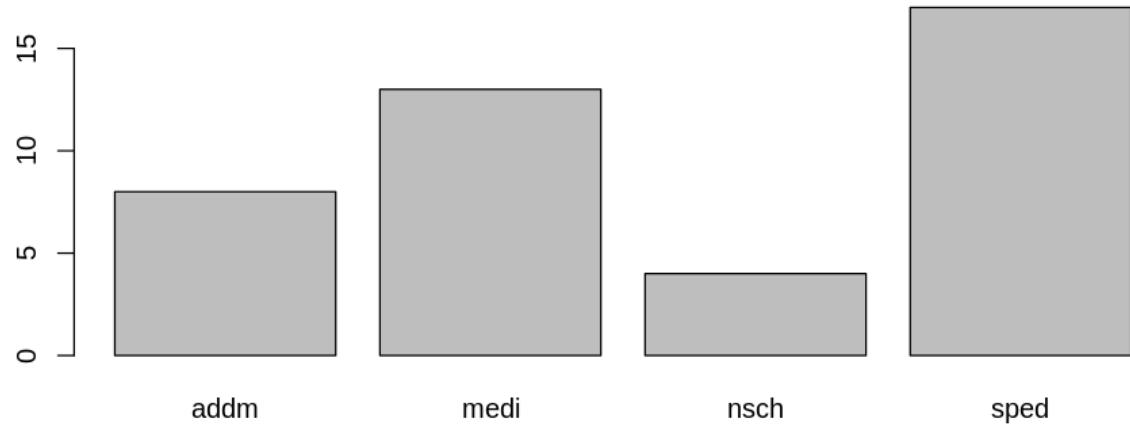
```
# -----  
# [National] < Years Data Available >  
# -----  
p = ggplot(ASD_National, aes(x = 1, fill = Source)) +  
    geom_bar() + theme(axis.text.x=element_blank(), # Hide axis  
                        axis.ticks.x=element_blank(), # Hide axis  
                        axis.text.y=element_blank(), # Hide axis  
                        axis.ticks.y=element_blank(), # Hide axis  
                        panel.background = element_blank(), # Remove panel background  
                        legend.position="top")  
) +  
    scale_fill_manual("Data Source:", values = c("addm" = "darkblue",  
                                                "medi" = "orange",  
                                                "nsch" = "darkred",  
                                                "sped" = "skyblue")) +  
    labs(x="", y="", title="Years Data Available") + # layers of graphics  
    facet_grid(facets = Source~Year)  
# Show plot  
p
```

Years Data Available

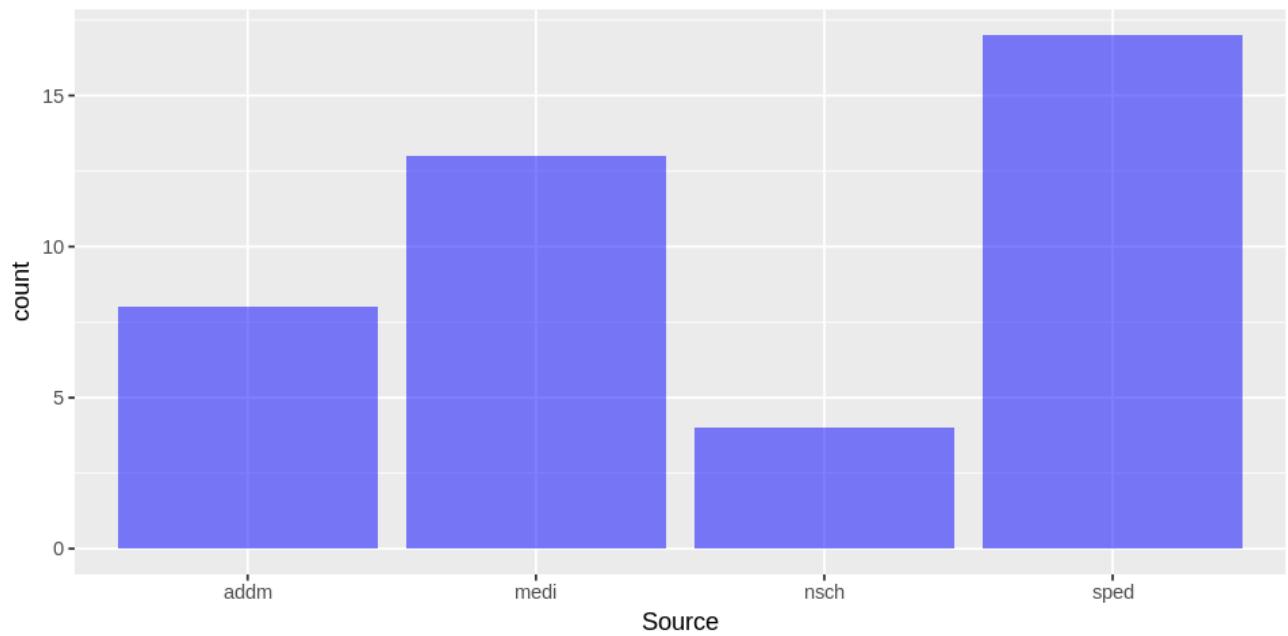


Data Visualisation (Enhanced) - Barplot

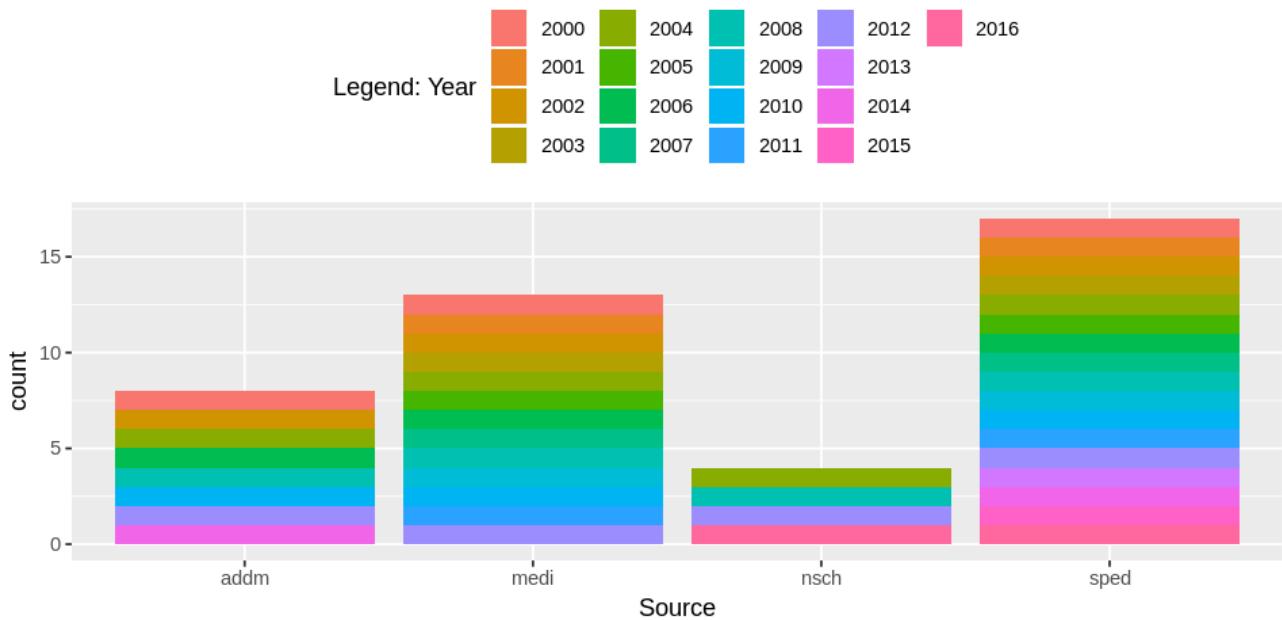
```
In [136]: # Create bar chart using R graphics  
barplot(table(ASD_National$Source))
```



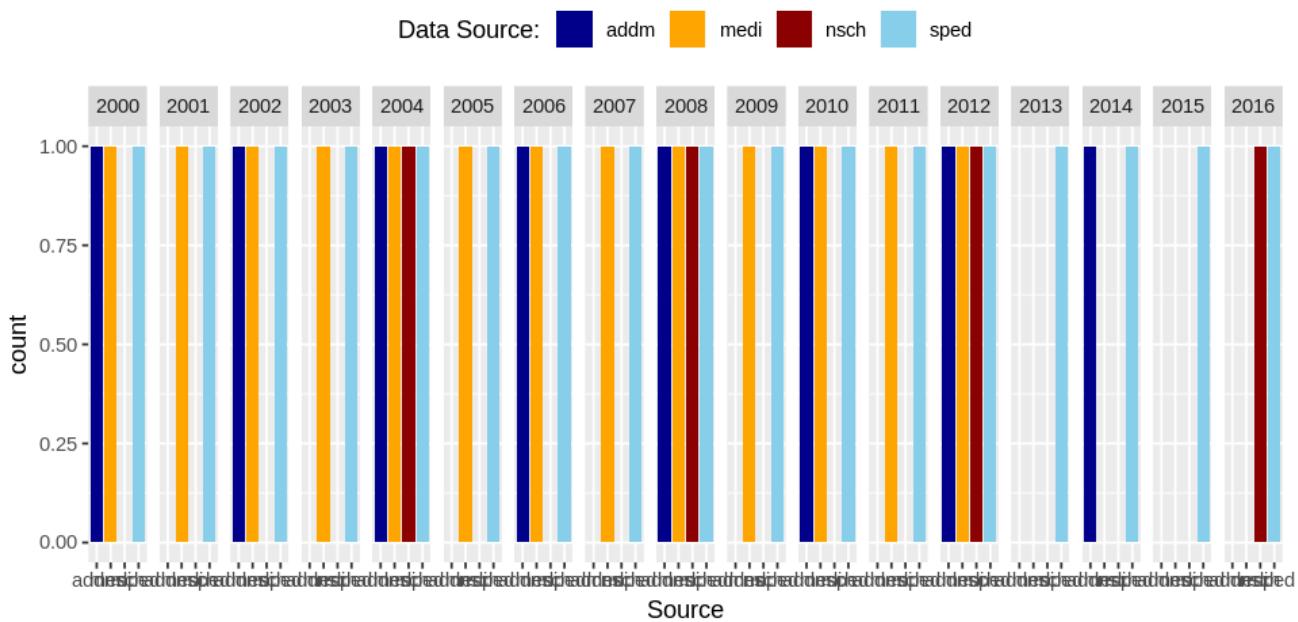
```
In [137]: # Create bar chart using ggplot2  
ggplot(ASD_National, aes(x = Source)) + geom_bar(fill = "blue", alpha=0.5)
```



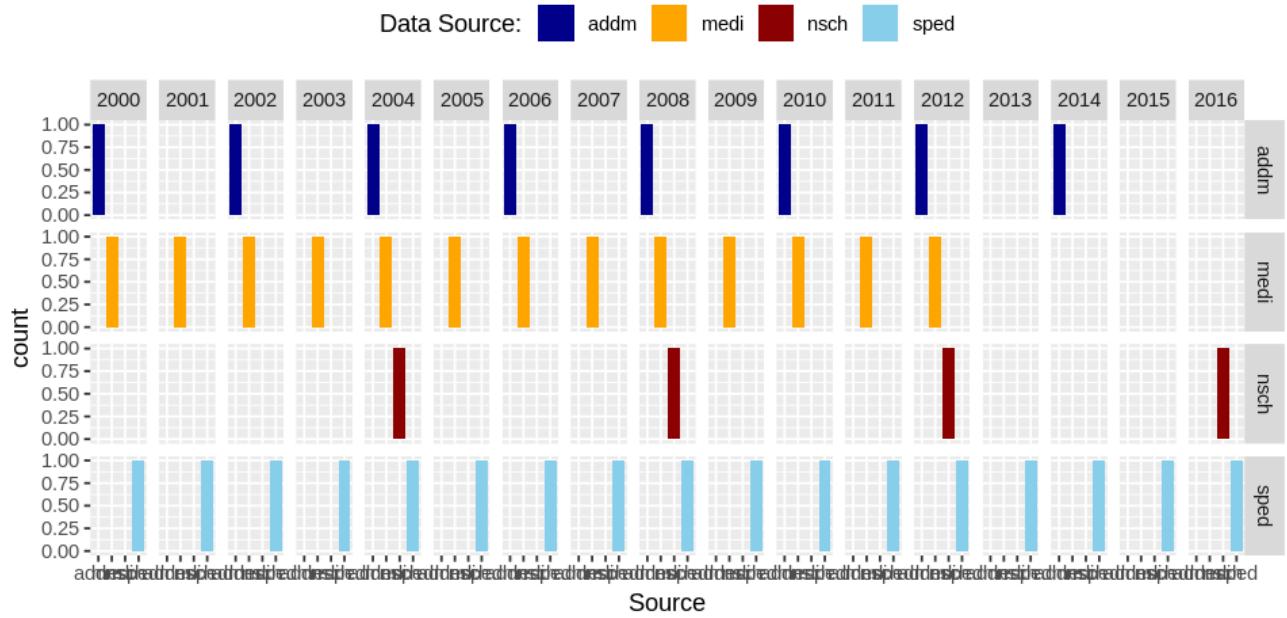
```
In [138]: # Use color to differentiate sub-group data (Year)
ggplot(ASD_National, aes(x = Source, fill = factor(Year))) + geom_bar() +
    theme(legend.position="top") + labs(fill = "Legend: Year")
```



```
In [139]: # Split chart to multiple columns by using: facets = . ~ Year
ggplot(ASD_National, aes(x = Source, fill = Source)) + geom_bar() +
  theme(legend.position="top") +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  facet_grid(facets = . ~ Year)
```



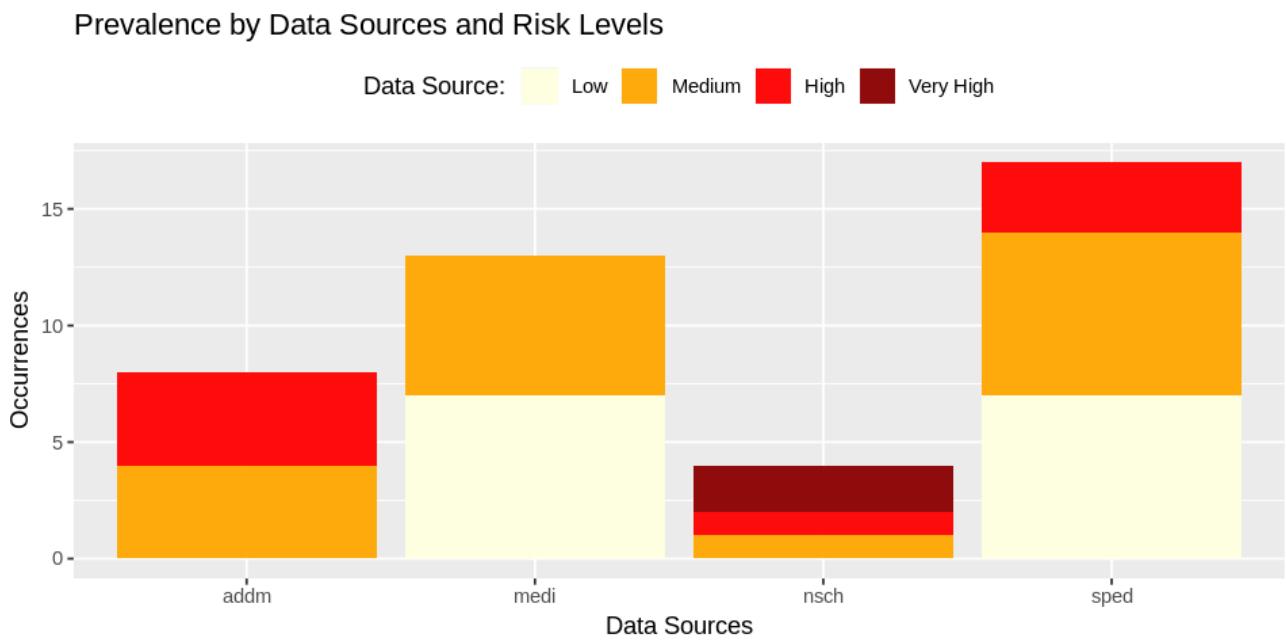
```
In [140]: # Split chart to multiple rows and columns by using: facets = Source ~ Year
ggplot(ASD_National, aes(x = Source, fill = Source)) + geom_bar() +
  theme(legend.position="top") +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  facet_grid(facets = Source~Year)
```



Above chart is now very similar to earlier [National] < Years Data Available >.

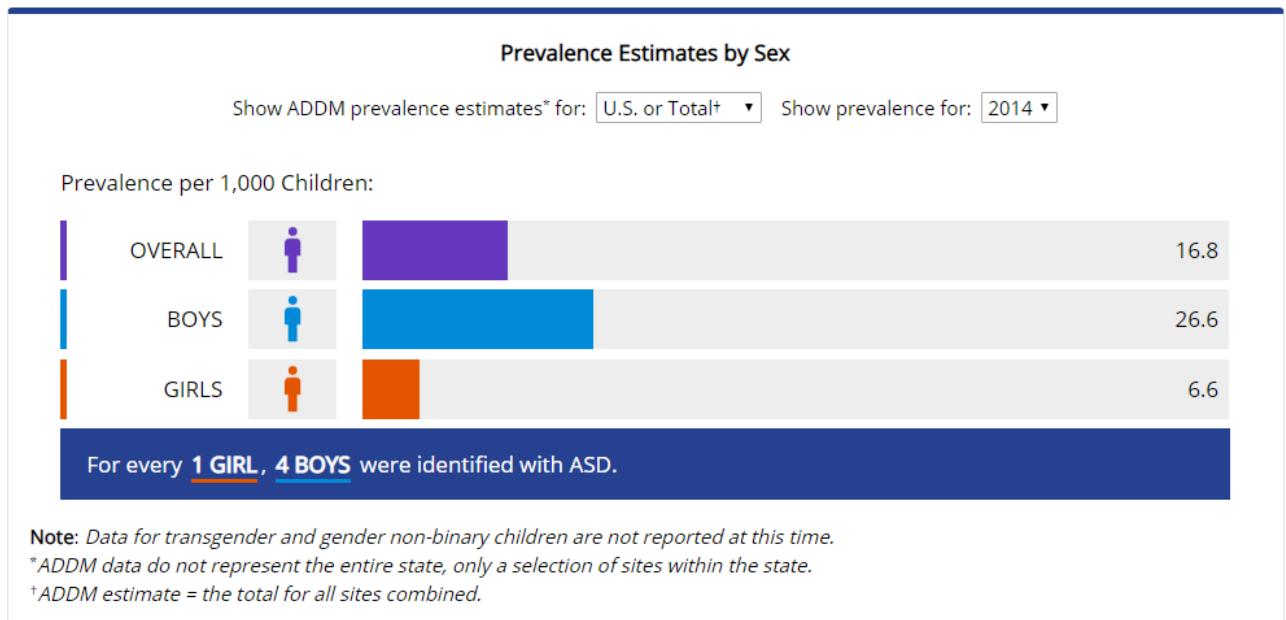
Data Visualisation (Enhanced) - [R] Prevalence by Data Sources and Risk Levels

```
In [141]: # Use color to differentiate sub-group data (Year)
ggplot(ASD_National, aes(x = Source, fill = Prevalence_Risk4)) +
  geom_bar(alpha=0.95, position = position_stack(reverse = TRUE)) + # Reverse
  scale_fill_manual("Data Source:", values = c("Low" = "lightyellow",
                                              "Medium" = "orange",
                                              "High" = "red",
                                              "Very High" = "darkred")) +
  labs(x="Data Sources", y="Occurrences", title="Prevalence by Data Sources an
theme(legend.position="top") + labs(fill = "Legend: Risk")
```



Barplot / Column plot

Data Visualisation (Enhanced) - [CDC] REPORTED PREVALENCE VARIES BY SEX



Data Visualisation (Enhanced) - [R] REPORTED PREVALENCE VARIES BY SEX [Source: ADDM] [Year: 2014]

In [142]: # Filter only data of ADDM

```
ASD_National_ADDM <- subset(ASD_National, Source == 'addm')
#
ASD_National_ADDM
```

Source	Year	Prevalence	Upper.Cl	Lower.Cl	Source_Full1	Source_Full2	Male.Prevalence	Male.Low
addm	2000	6.7	7.0	6.3	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	NA	
addm	2002	6.6	6.8	6.3	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	11.5	
addm	2004	8.0	8.4	7.6	Autism & Developmental Disabilities Monitoring Network	addm-Autism & Developmental Disabilities Monitoring Network	12.9	

In [143]: # Construct a new re-shaped dataframe of [Source: ADDM] [Year: 2014]

```
#  
Process_Source = 'addm'  
Process_Year = 2014
```

Define a function to create a re-shaped dataframe:

```
In [144]: Function_Reshape_ASDD_National_ADDM <- function(Process_Source, Process_Year) {  
  # Create the vectors:  
  Sex.Group = c('Overall',  
              'Boys',  
              'Girls')  
  Sex.Group  
  
  Prevalence = c(ASD_National_ADDM$Prevalence[ASD_National_ADDM$Year == Proc  
ASD_National_ADDM$Male.Prevalence[ASD_National_ADDM$Year ==  
ASD_National_ADDM$Female.Prevalence[ASD_National_ADDM$Year  
Prevalence  
  
  # Combine all the vectors into a data frame:  
  ASD_National_ADDM_Reshaped_DF = data.frame(Sex.Group, Prevalence, stringsAsFactors = FALSE)  
  
  # Add new columns:  
  ASD_National_ADDM_Reshaped_DF$Source = Process_Source  
  ASD_National_ADDM_Reshaped_DF$Year = Process_Year  
  return(ASD_National_ADDM_Reshaped_DF) # Return a dataframe  
}
```

Use defined function **Function_Reshape_ASDD_National_ADDM()** for a specific year:

```
In [145]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASDD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF
```

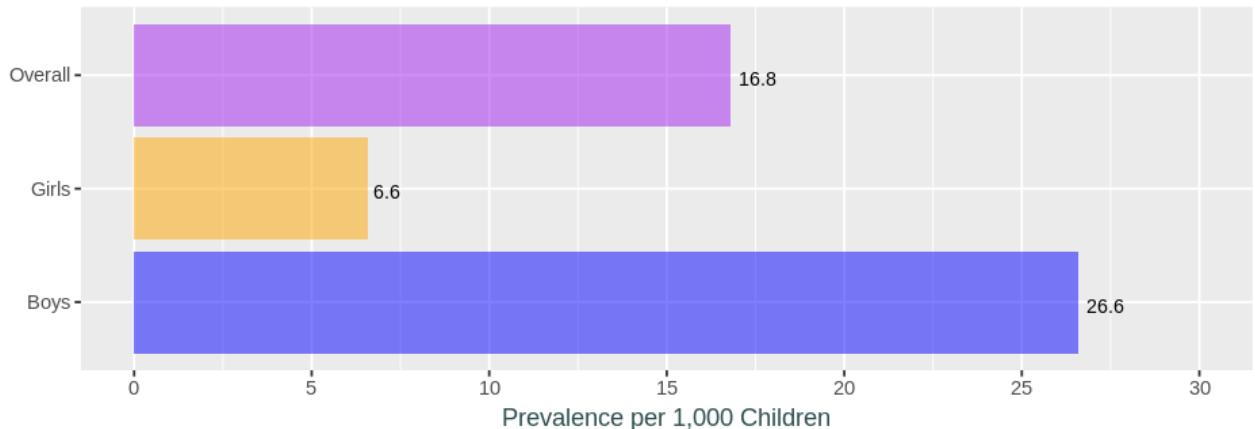
Sex.Group	Prevalence	Source	Year
Overall	16.8	addm	2014
Boys	26.6	addm	2014
Girls	6.6	addm	2014

Visualise: Prevalence Estimates by Sex [Source: ADDM] [Year: 2014]

```
In [146]: # Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=3)
```

```
In [147]: ggplot(ASD_National_ADDM_Reshaped_DF, aes(Sex.Group, Prevalence)) +  
  geom_col(aes(fill = Sex.Group, colours = ), alpha=0.5) + # Use column chart  
  geom_text(aes(label = Prevalence), vjust = +0.75, hjust = -0.2, size = 3) +  
  scale_y_continuous(name = "Prevalence per 1,000 Children",  
                     breaks = seq(0, 30, 5),  
                     limits=c(0, 30)) +  
  scale_x_discrete(name = "") +  
  scale_fill_manual("Sex Group:", values = c("Overall" = "purple",  
                                         "Boys" = "blue",  
                                         "Girls" = "orange")) +  
  ggtile("Prevalence Estimates by Sex [ Source: ADDM ] [ Year: 2014 ]") +  
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),  
        axis.title = element_text(face = 'plain', color = "darkslategrey"),  
        legend.position = 'none') +  
  coord_flip() # Rotate chart  
# facet_grid(facets = Year ~ .)
```

Prevalence Estimates by Sex [Source: ADDM] [Year: 2014]



Data Visualisation (Enhanced) - [R] REPORTED PREVALENCE VARIES BY SEX [Source: ADDM] [Year: ALL]

```
In [148]: # Create a new datafarme to hold re-shaped data for all years.  
ASD_National_ADDM_Reshaped_DF_All = ASD_National_ADDM_Reshaped_DF # Loaded wit
```

```
In [149]: Process_Source = 'addm'  
unique(ASD_National_ADDM$Year)
```

2000 2002 2004 2006 2008 2010 2012 2014

Use defined function `Function_Reshape_ASD_National_ADDM()` for ALL remaining years:

```
In [150]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	14.8	addm	2012
Boys	23.4	addm	2012
Girls	5.2	addm	2012

```
In [151]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	14.7	addm	2010
Boys	23.7	addm	2010
Girls	5.3	addm	2010

```
In [152]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	11.3	addm	2008
Boys	18.4	addm	2008
Girls	4.0	addm	2008

```
In [153]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	9.0	addm	2006
Boys	14.5	addm	2006
Girls	3.2	addm	2006

```
In [154]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	8.0	addm	2004
Boys	12.9	addm	2004
Girls	2.9	addm	2004

```
In [155]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASDD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	6.6	addm	2002
Boys	11.5	addm	2002
Girls	2.7	addm	2002

```
In [156]: ASD_National_ADDM_Reshaped_DF <- Function_Reshape_ASDD_National_ADDM(Process_Sc  
ASD_National_ADDM_Reshaped_DF  
# Append rows to existing dataframe, using Row Bind function: rbind()  
ASD_National_ADDM_Reshaped_DF_All = rbind(ASD_National_ADDM_Reshaped_DF_All, A
```

Sex.Group	Prevalence	Source	Year
Overall	6.7	addm	2000
Boys	NA	addm	2000
Girls	NA	addm	2000

```
In [157]: # Re-shaped ADDM data for ALL years:  
ASD_National_ADDM_Reshaped_DF_All
```

Sex.Group	Prevalence	Source	Year
Overall	16.8	addm	2014
Boys	26.6	addm	2014
Girls	6.6	addm	2014
Overall	14.8	addm	2012
Boys	23.4	addm	2012
Girls	5.2	addm	2012
Overall	14.7	addm	2010
Boys	23.7	addm	2010
Girls	5.3	addm	2010
Overall	11.3	addm	2008
Boys	18.4	addm	2008
Girls	4.0	addm	2008

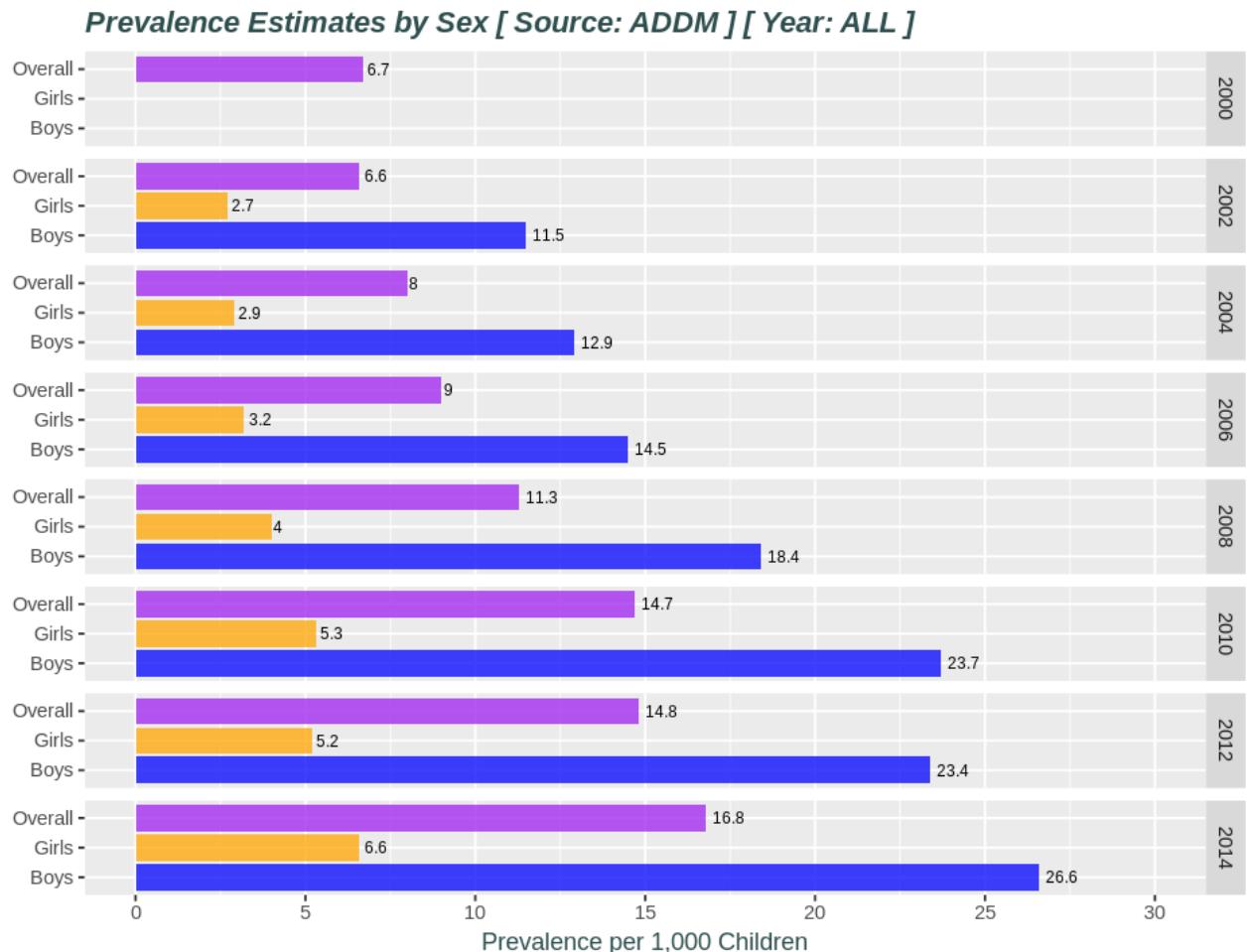
Visualise: Prevalence Estimates by Sex [Source: ADDM] [Year: ALL]

```
In [158]: # Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=6)
```

```
In [159]: ggplot(ASD_National_ADDM_Reshaped_DF_All, aes(Sex.Group, Prevalence)) +
  geom_col(aes(fill = Sex.Group, colours = ), alpha=0.75) + # Use column chart
  geom_text(aes(label = Prevalence), vjust = +0.5, hjust = -0.2, size = 2.5) +
  scale_y_continuous(name = "Prevalence per 1,000 Children",
                     breaks = seq(0, 30, 5),
                     limits=c(0, 30)) +
  scale_x_discrete(name = "") +
  scale_fill_manual("Sex Group:", values = c("Overall" = "purple",
                                             "Boys" = "blue",
                                             "Girls" = "orange")) +
  ggtile("Prevalence Estimates by Sex [ Source: ADDM ] [ Year: ALL ]") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"),
        legend.position = 'none') +
  coord_flip() + # Rotate chart
  facet_grid(facets = Year ~ .)
```

Warning message:

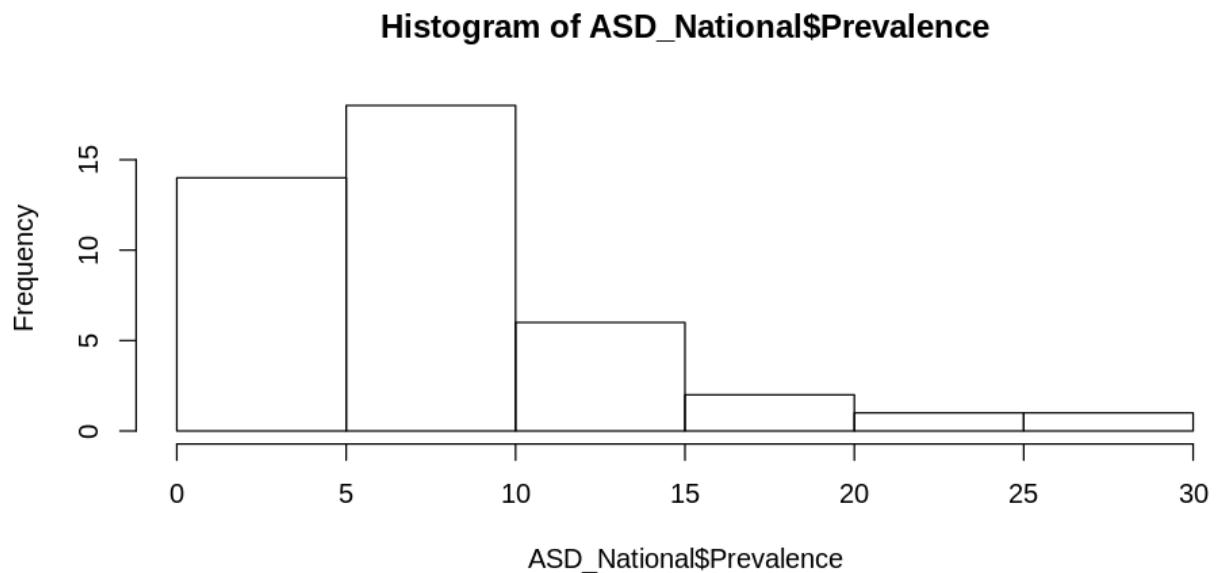
"Removed 2 rows containing missing values (position_stack)." Warning message:
"Removed 2 rows containing missing values (geom_text)."



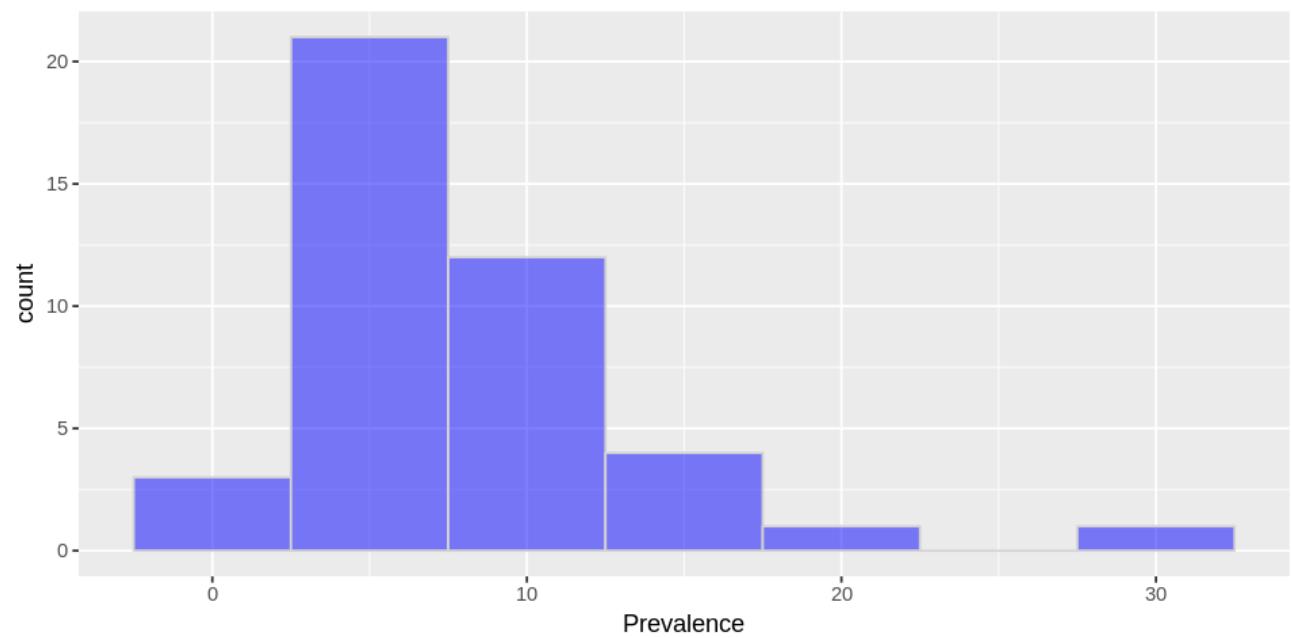
Data Visualisation (Enhanced) - Histogram (distribution of binned continuous variable)

```
In [160]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)
```

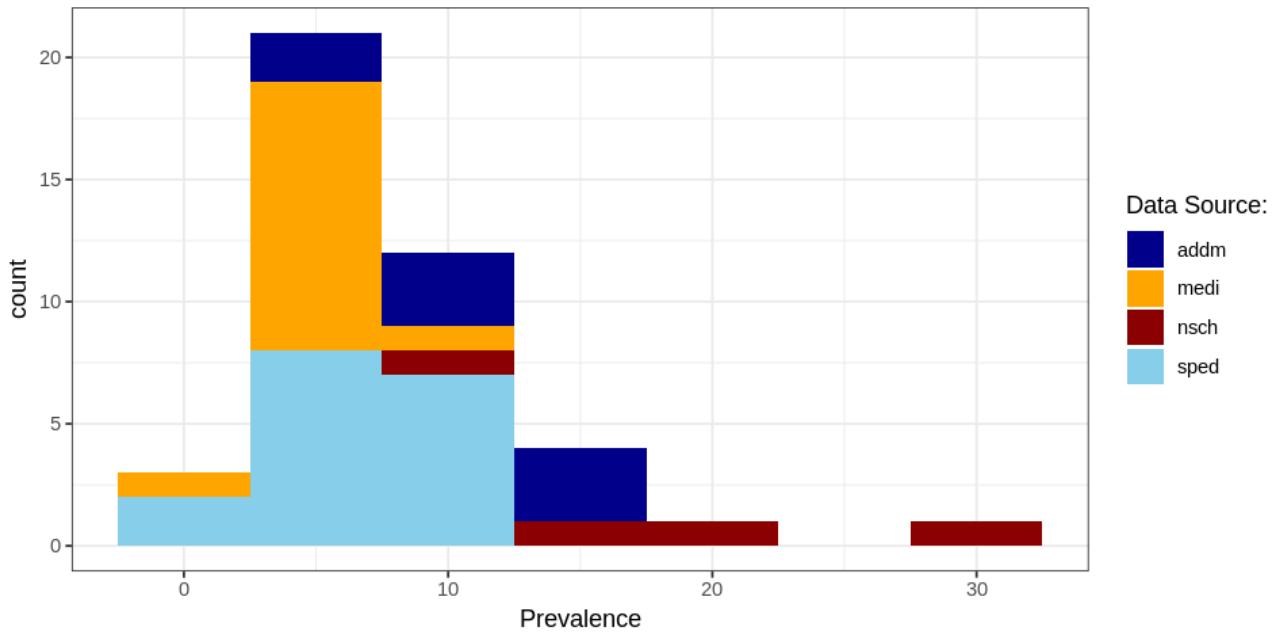
```
In [161]: # Create histogram using R graphics  
hist(ASD_National$Prevalence)
```



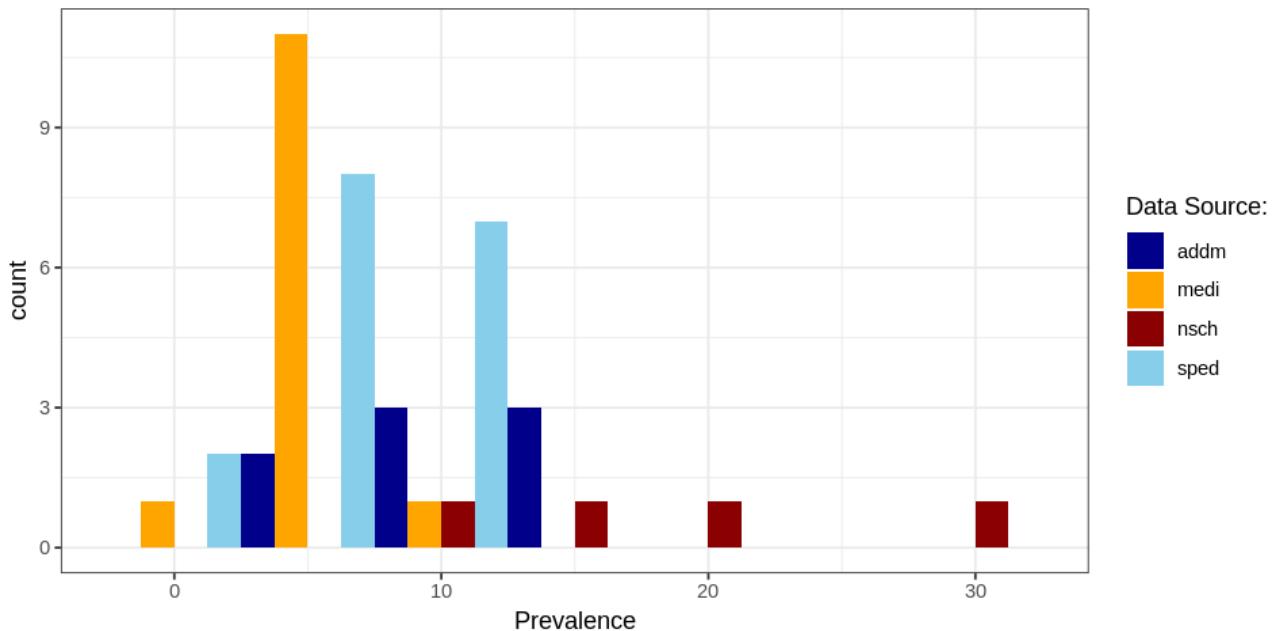
```
In [162]: # Create histogram using ggplot2  
ggplot(ASD_National, aes(x=Prevalence)) +  
  geom_histogram(binwidth = 5, fill = "blue", color = "lightgrey", alpha=0.5)
```



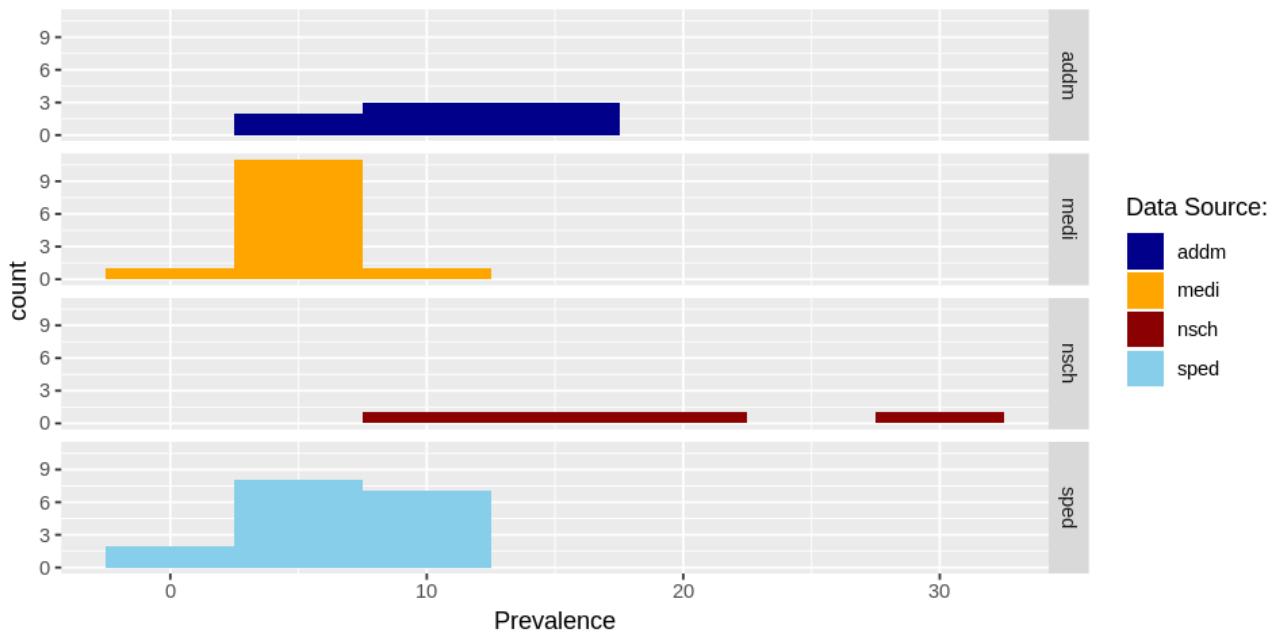
```
In [163]: # Use color to differentiate sub-group data (Data Source)
ggplot(ASD_National, aes(x=Prevalence, fill = Source)) +
  geom_histogram(binwidth = 5) +
  theme_bw() + theme(legend.position="right") +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue"))
```



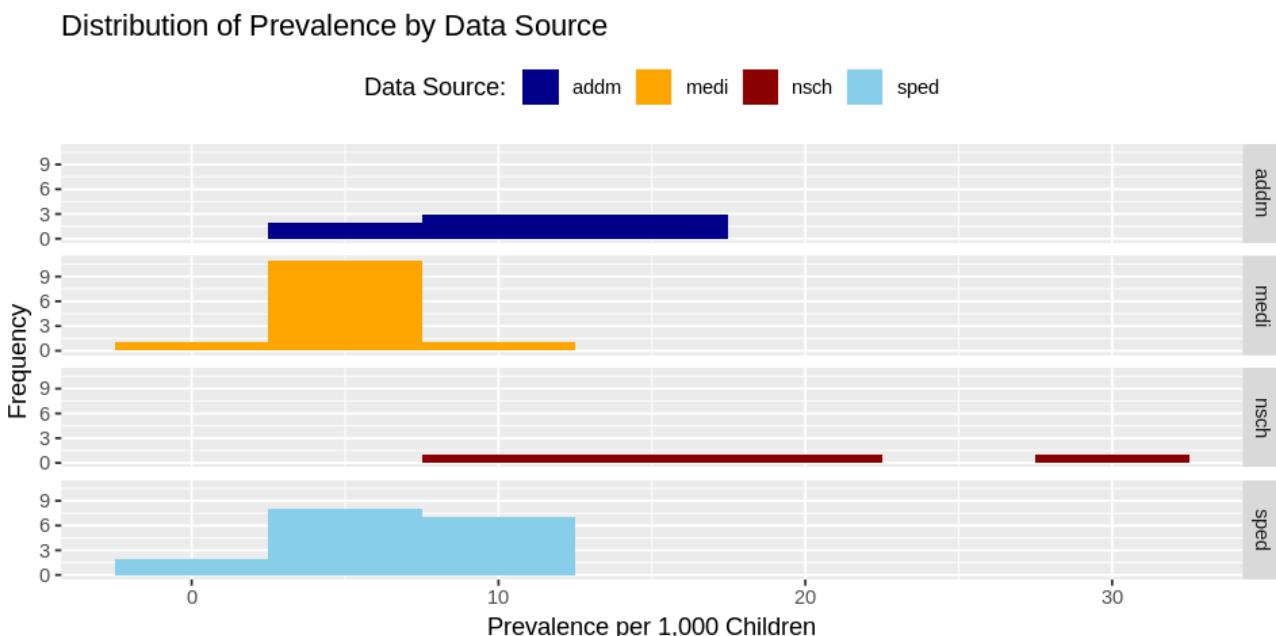
```
In [164]: # Plot sub-group data side by side, using position="dodge"
ggplot(ASD_National, aes(x=Prevalence, fill = Source)) +
  geom_histogram(binwidth = 5, position="dodge") +
  theme_bw() + theme(legend.position="right") +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue"))
```



```
In [165]: # Split plots using facet_grid()
ggplot(ASD_National, aes(x=Prevalence, fill = Source)) +
  geom_histogram(binwidth = 5) +
  theme(legend.position="right") +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  facet_grid(facets = Source ~ .)
```



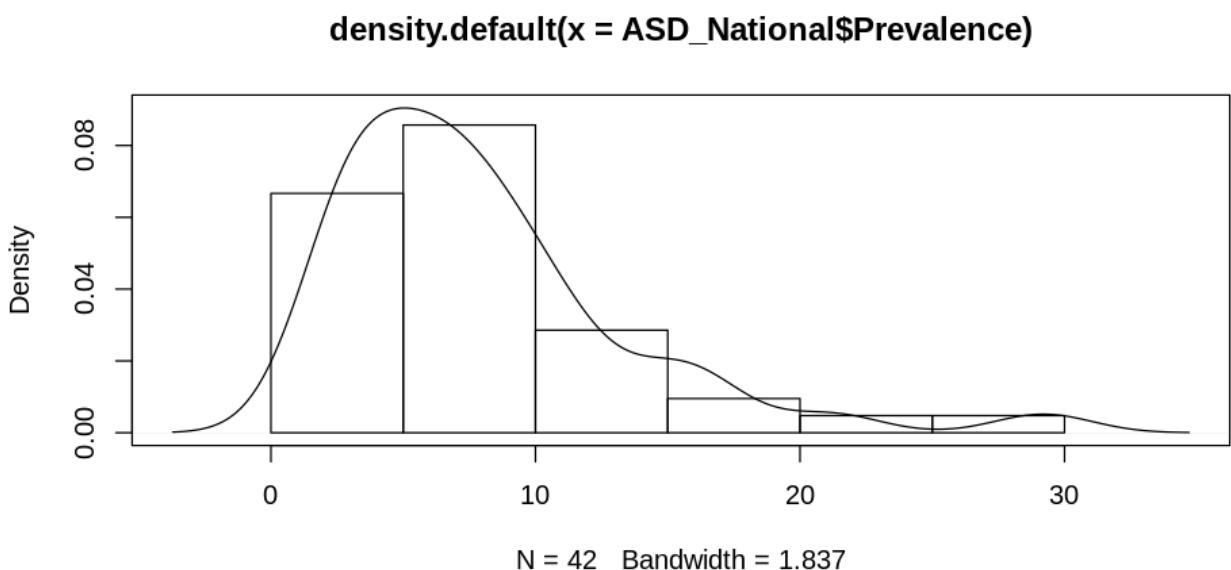
```
In [166]: # Add title and caption using ggplot2
ggplot(ASD_National, aes(x=Prevalence, fill = Source)) +
  geom_histogram(binwidth = 5) +
  theme(legend.position="top") +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  labs(x="Prevalence per 1,000 Children",
       y="Frequency",
       title="Distribution of Prevalence by Data Source") +
  facet_grid(facets = Source ~ .)
```



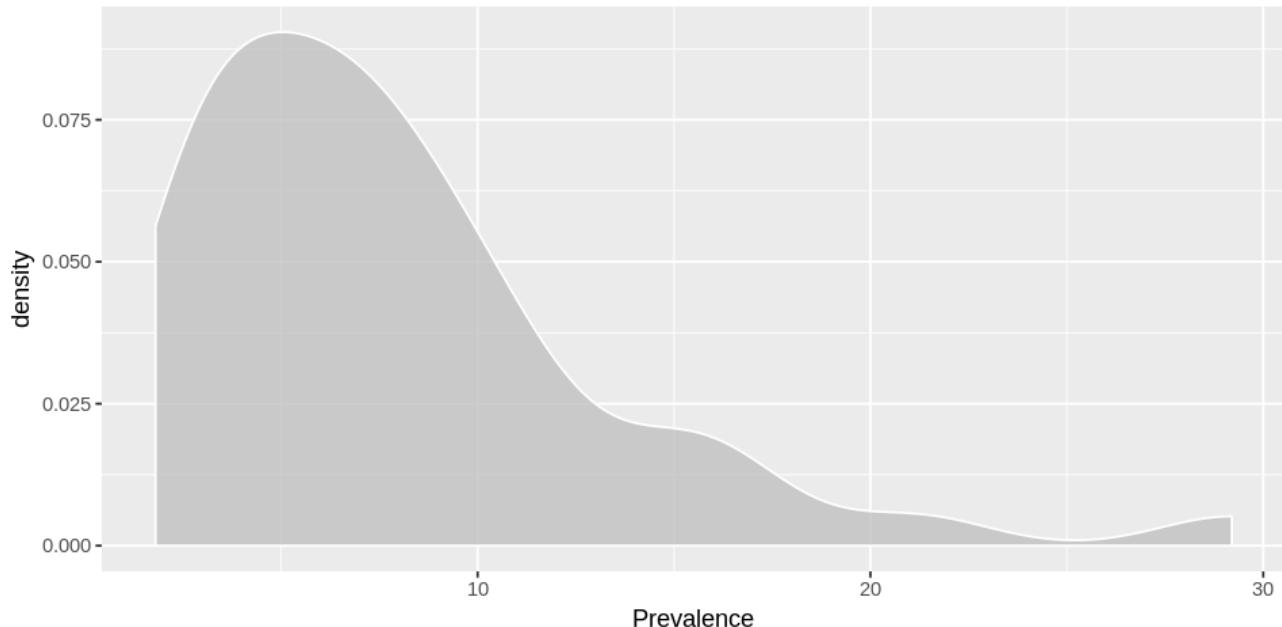
Data Visualisation (Enhanced) - Density plot (distribution for continuous variable normalized to 100% area under curve)

```
In [167]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

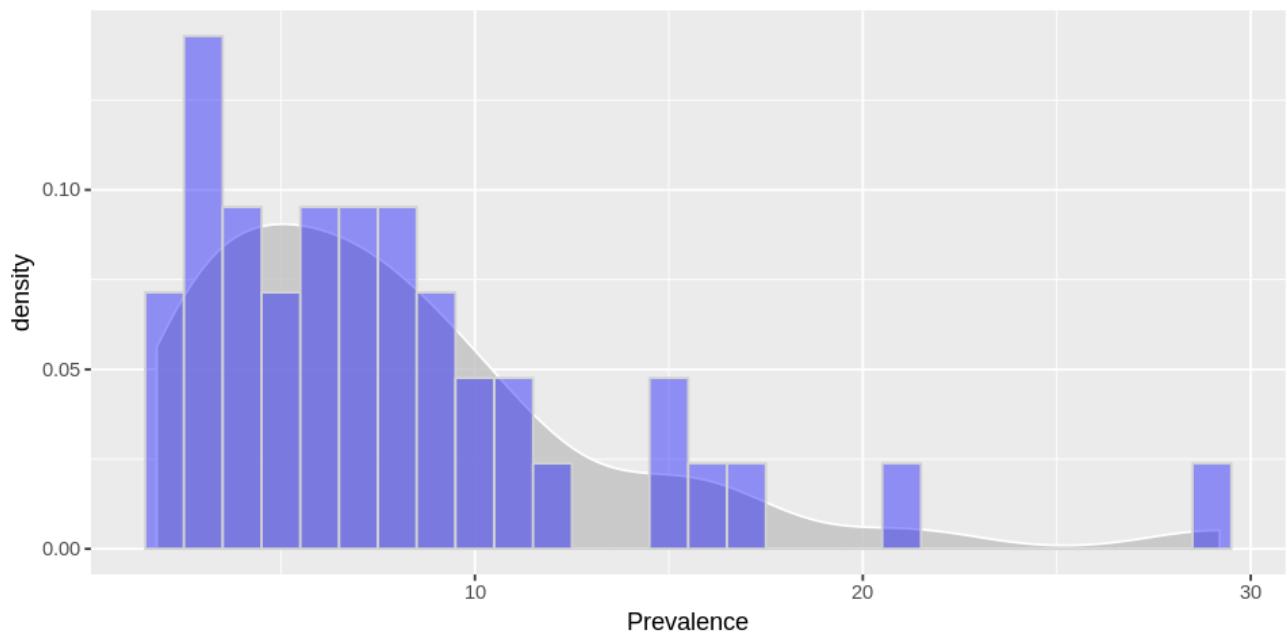
```
In [168]: # Create plot using R graphics  
plot(density(ASD_National$Prevalence))  
# Optionally, overlay histogram  
hist(ASD_National$Prevalence, probability = TRUE, add = TRUE)
```



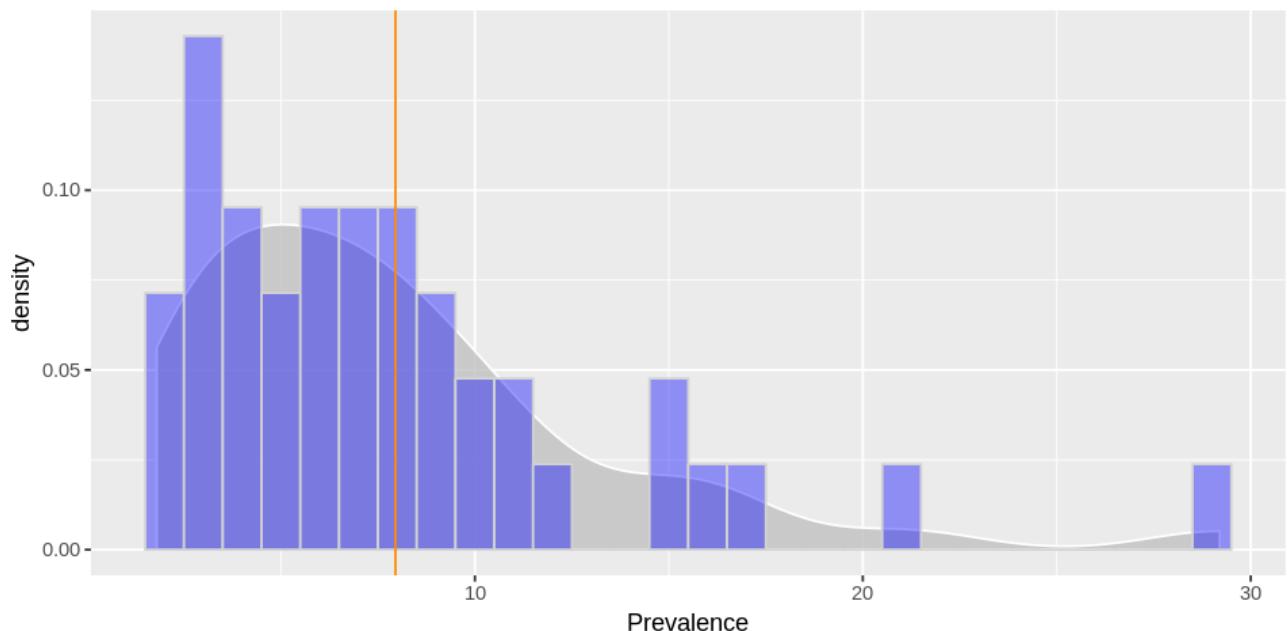
```
In [169]: # Create plot using ggplot2  
p <- ggplot(ASD_National) +  
  geom_density(aes(x=Prevalence), fill = "grey", color = "white", alpha=0.75)  
p # Show
```



```
In [170]: # Optionally, overlay histogram  
p <- p + geom_histogram(aes(x = Prevalence, y = ..density..), binwidth = 1, fi  
p # Show
```

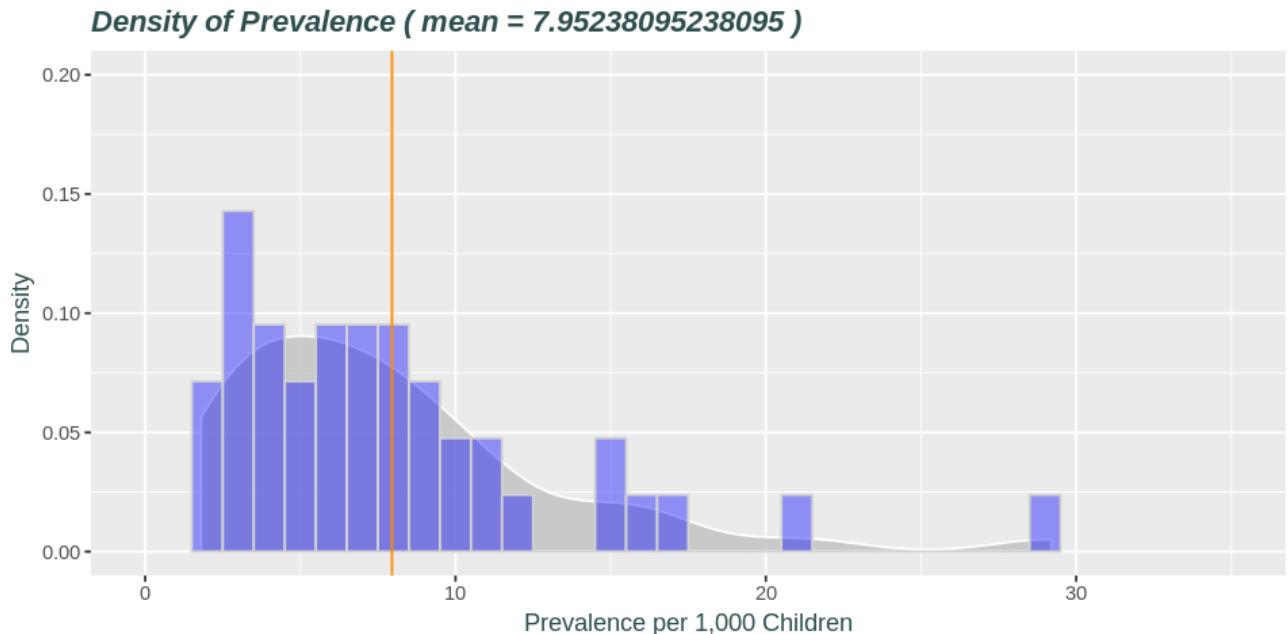


```
In [171]: # Optionally, overlay Prevalence mean  
p <- p + geom_vline(aes(xintercept = mean(ASD_National$Prevalence)), colour="d  
p # Show
```



In [172]: # Lastly, add other captions

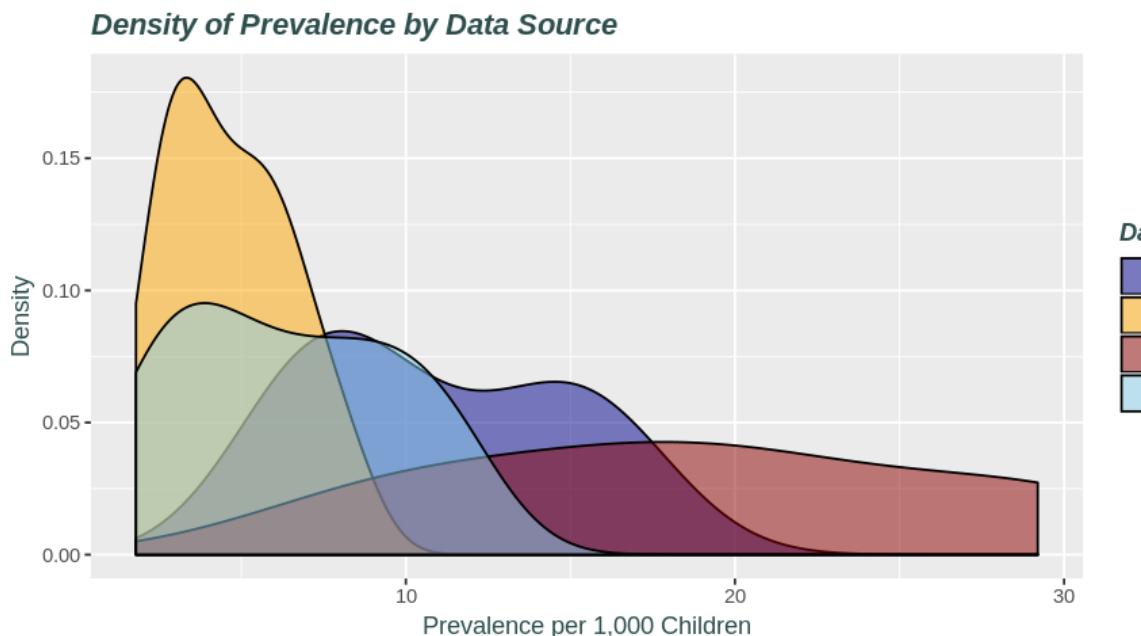
```
p <- p + coord_cartesian(xlim=c(0, 35), ylim=c(0, 0.2)) +
  labs(x="Prevalence per 1,000 Children", y="Density",
       title= paste("Density of Prevalence ( mean =", mean(ASD_National$Prevalence),
       theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
             axis.title = element_text(face = 'plain', color = "darkslategrey")))
p # Show
```



< Prevelance distribution by Data Source >

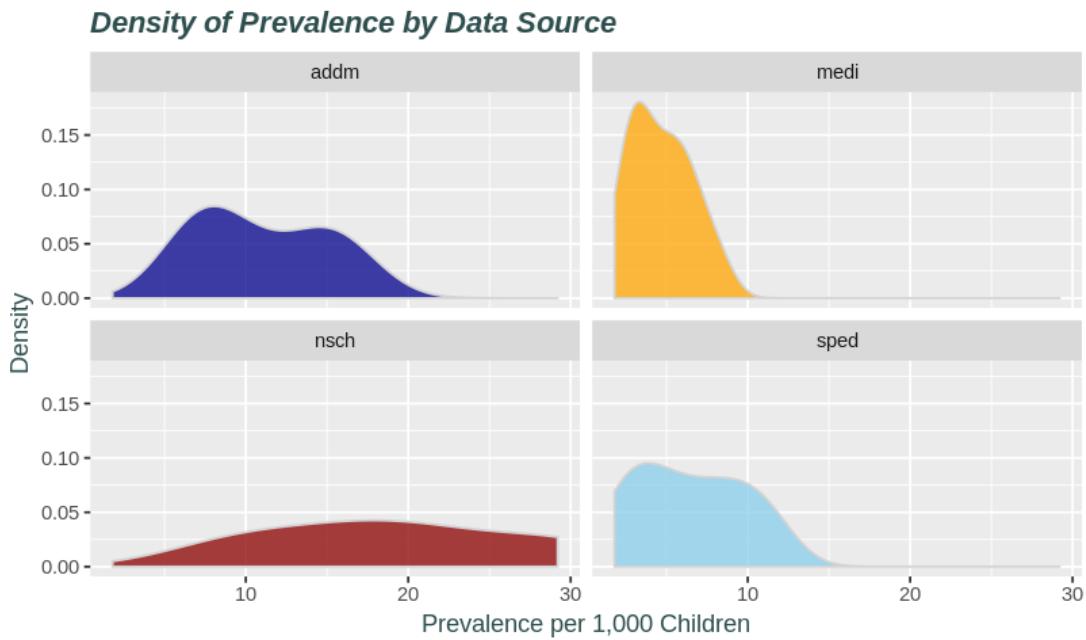
In [173]: # Prevelance distribution by Data Source

```
ggplot(ASD_National) + geom_density(aes(x = Prevalence, fill = Source), alpha =
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  labs(x="Prevalence per 1,000 Children",
       y="Density",
       title="Density of Prevalence by Data Source") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"))
```



< Prevelance distribution by Data Source with split >

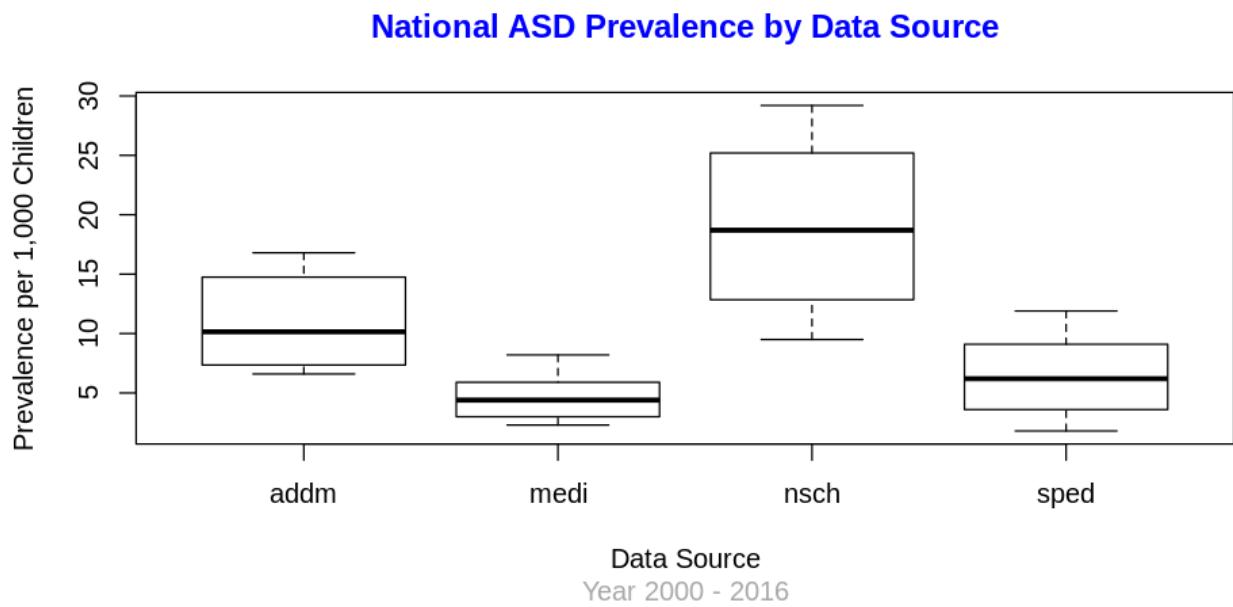
```
In [174]: # Prevelance distribution by Data Source with split
ggplot(ASD_National) + geom_density(aes(x = Prevalence, fill = Source), colour
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  labs(x="Prevalence per 1,000 Children",
       y="Density",
       title="Density of Prevalence by Data Source") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey")) +
  facet_wrap(~Source)
```



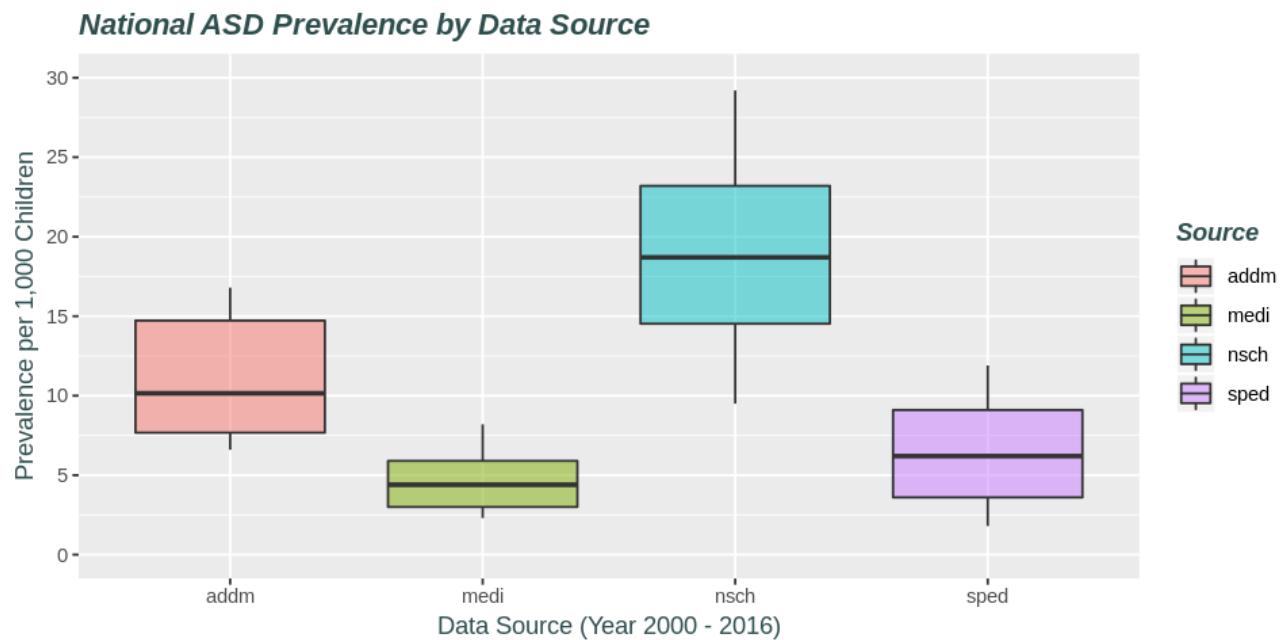
Data Visualisation (Enhanced) - Box plot

```
In [175]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)
```

```
In [176]: # Create plot using R graphics
# Create 'Prevalence' box plots break by 'Source'
boxplot(ASD_National$Prevalence ~ ASD_National$Source,
        main = "National ASD Prevalence by Data Source",
        xlab = "Data Source",
        ylab = "Prevalence per 1,000 Children",
        sub = "Year 2000 - 2016",
        col.main="blue", col.lab="black", col.sub="darkgrey")
```



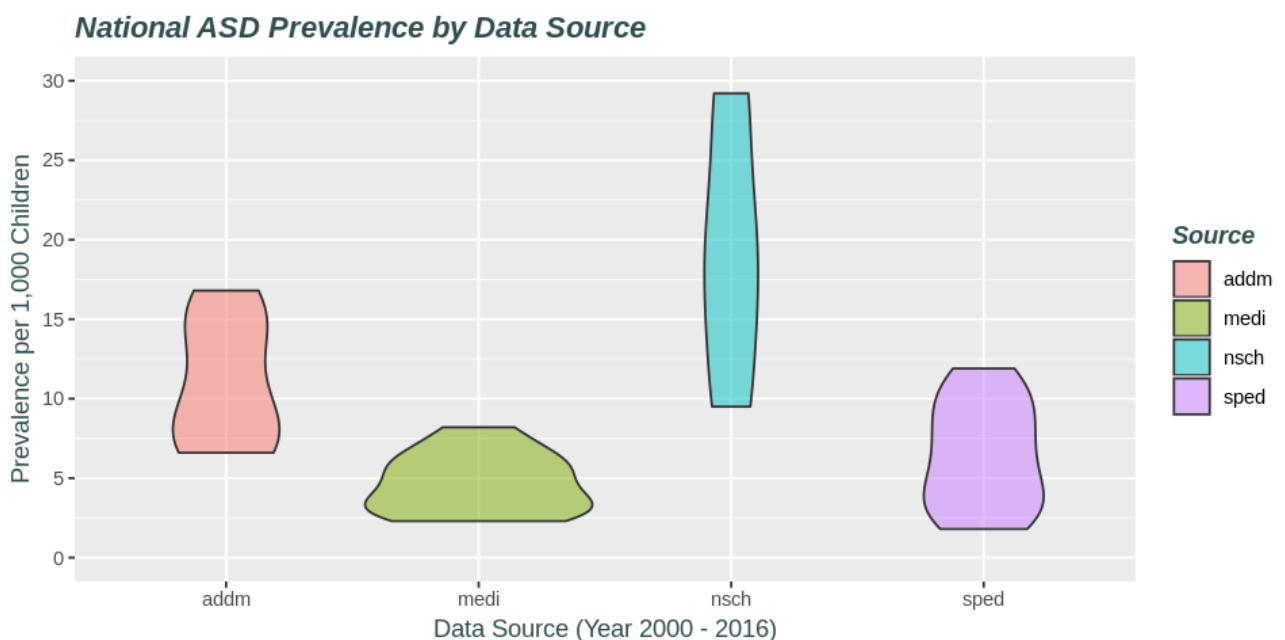
```
In [177]: # Create box plot using ggplot2
ggplot(ASD_National, aes(x = Source, y = Prevalence, fill = Source)) +
  geom_boxplot(alpha = 0.5) +
  scale_y_continuous(name = "Prevalence per 1,000 Children",
                     breaks = seq(0, 30, 5),
                     limits=c(0, 30)) +
  scale_x_discrete(name = "Data Source (Year 2000 - 2016)") +
  ggtitle("National ASD Prevalence by Data Source") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"))
```



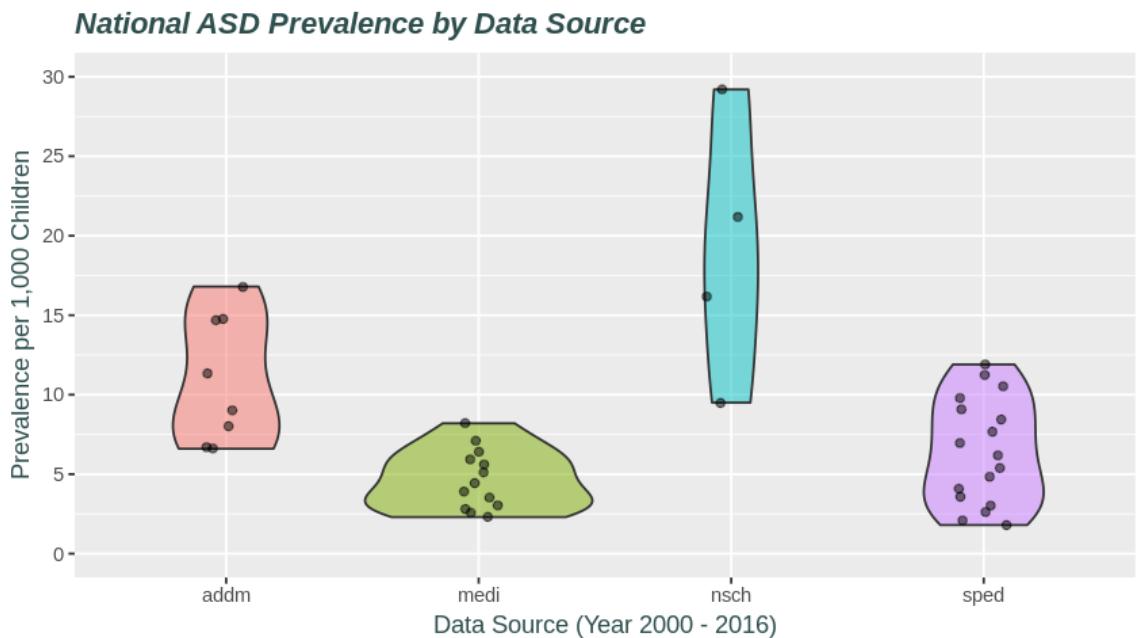
Data Visualisation (Enhanced) - Violin plot

```
In [178]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

```
In [179]: # Create plot using ggplot2  
ggplot(ASD_National, aes(x = Source, y = Prevalence, fill = Source)) +  
  geom_violin(alpha = 0.5) +  
  scale_y_continuous(name = "Prevalence per 1,000 Children",  
                     breaks = seq(0, 30, 5),  
                     limits=c(0, 30)) +  
  scale_x_discrete(name = "Data Source (Year 2000 - 2016)") +  
  ggtitle("National ASD Prevalence by Data Source") +  
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),  
        axis.title = element_text(face = 'plain', color = "darkslategrey"))
```

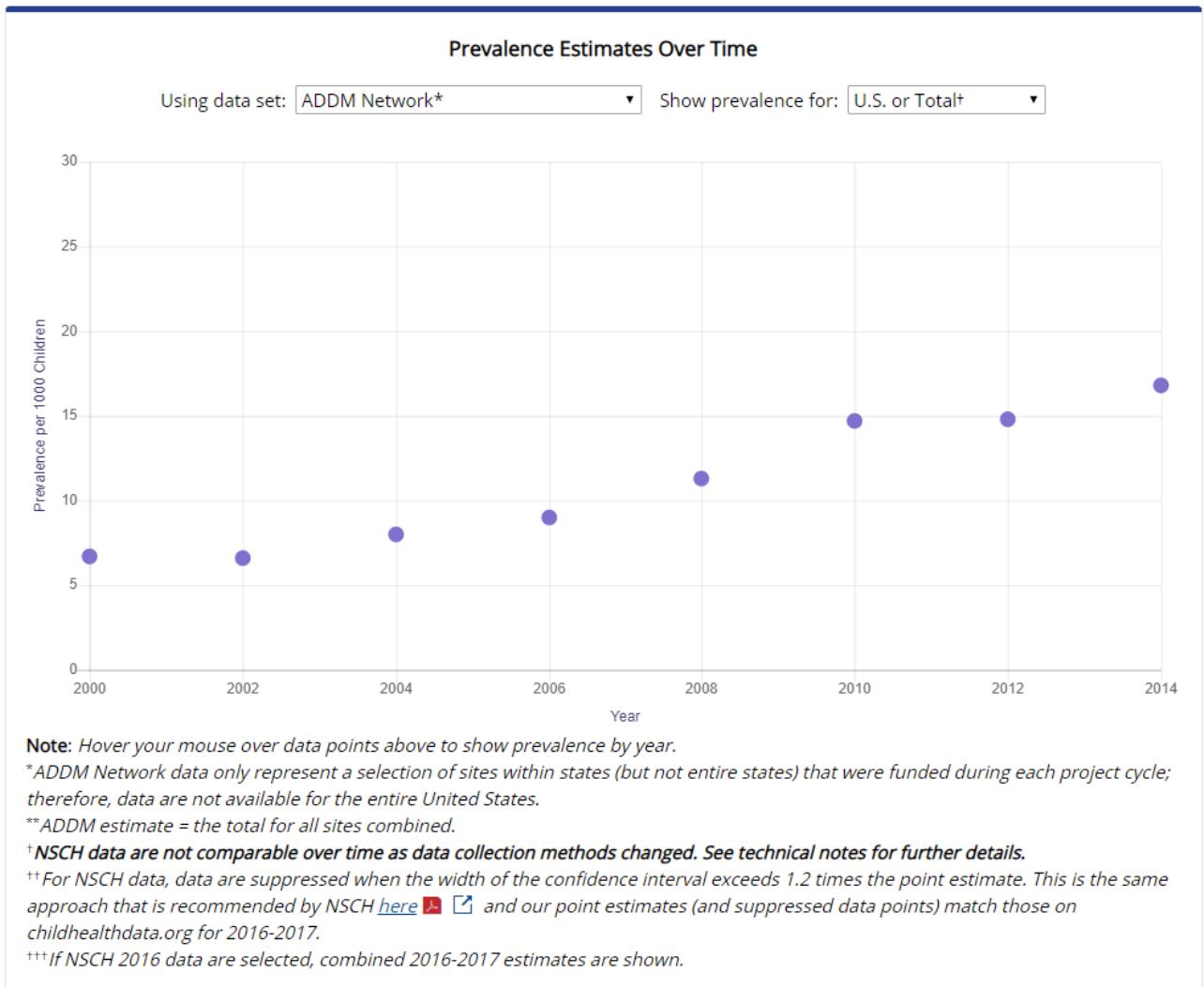


```
In [180]: # Create plot using ggplot2
ggplot(ASD_National, aes(x = Source, y = Prevalence, fill = Source)) +
  geom_violin(alpha = 0.5) +
  geom_jitter(alpha = 0.5, position = position_jitter(width = 0.1)) + # Overlap
# coord_flip() + # Uncomment to flip x-y axis
  scale_y_continuous(name = "Prevalence per 1,000 Children",
                     breaks = seq(0, 30, 5),
                     limits=c(0, 30)) +
  scale_x_discrete(name = "Data Source (Year 2000 - 2016)") +
  ggtitle("National ASD Prevalence by Data Source") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"))
```



Data Visualisation (Enhanced) - Line chart

Data Visualisation (Enhanced) - [CDC] REPORTED PREVALENCE HAS CHANGED OVER TIME

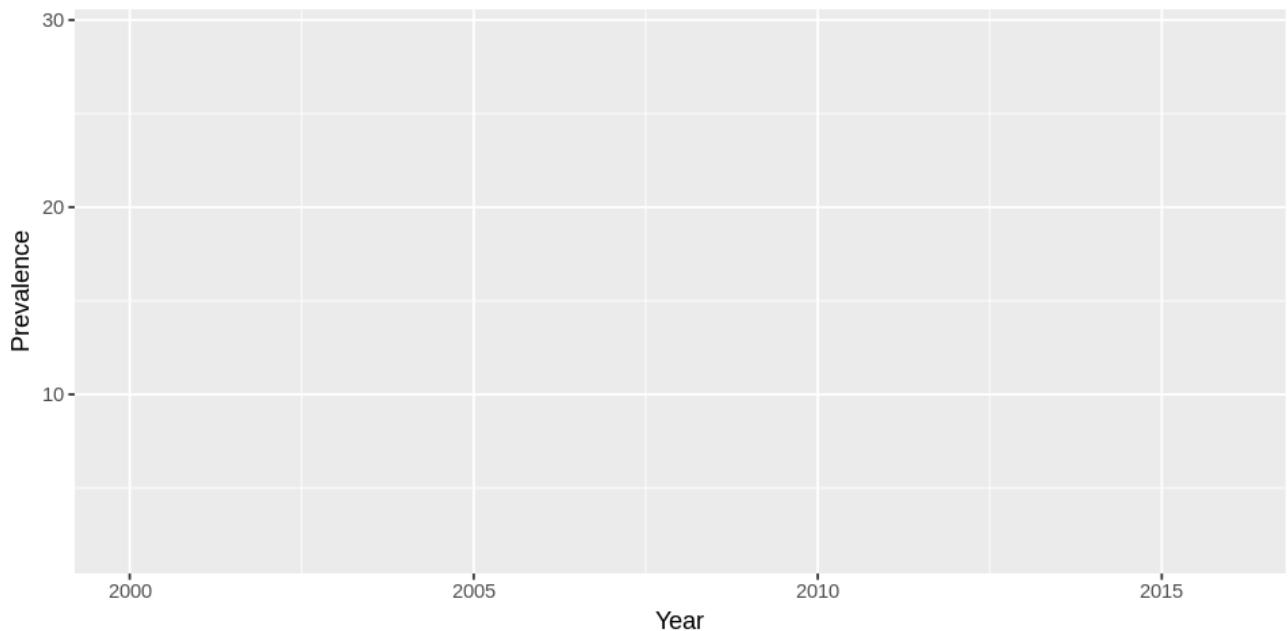


Data Visualisation (Enhanced) - [R] REPORTED PREVALENCE HAS CHANGED OVER TIME [Source: ALL]

```
In [181]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)
```

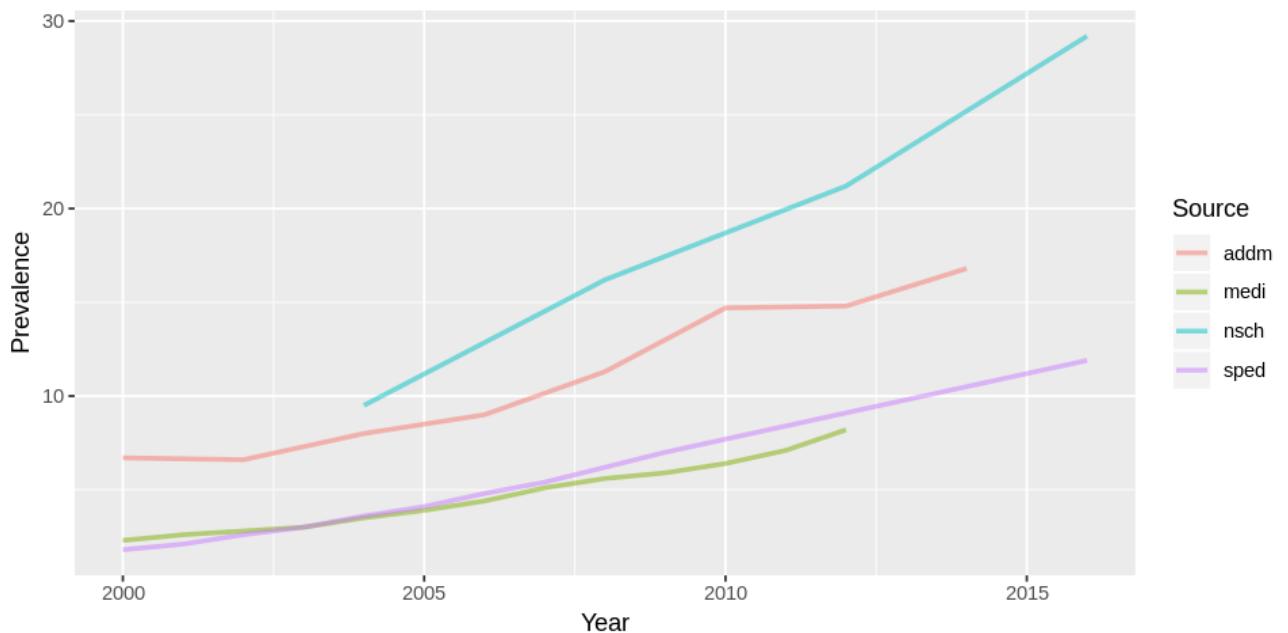
In [182]:

```
# -----  
# Build chart/plot layer by layer  
# -----  
  
# Define a ggplot graphic object; provide data and x y for use  
p <- ggplot(ASD_National, aes(x = Year, y = Prevalence))  
# Show plot  
p
```

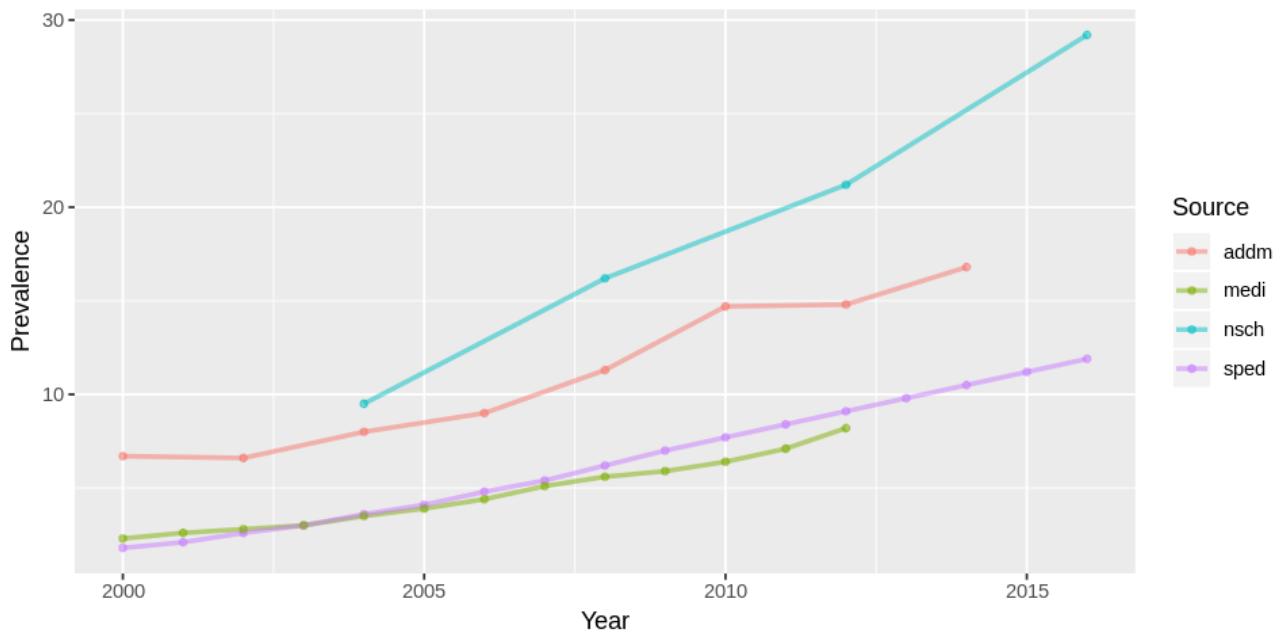


In [183]:

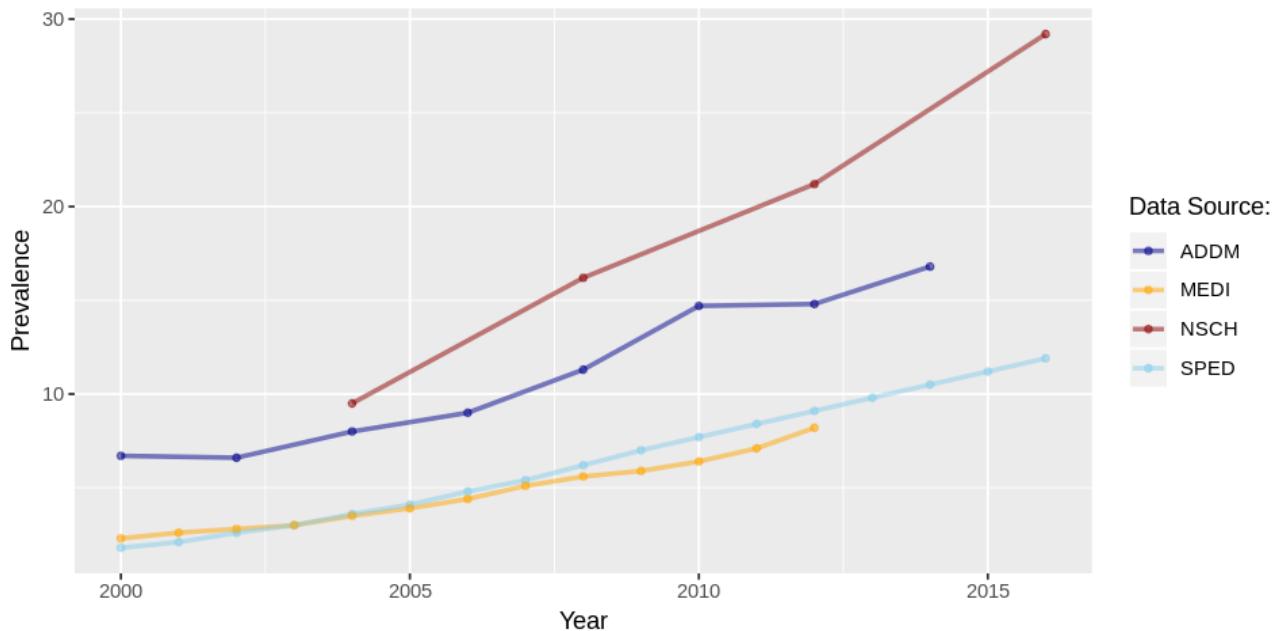
```
# Select (add) line chart type:  
p <- p + geom_line(aes(color = Source),  
                     linetype = "solid", # http://sape.inf.usi.ch/quick-referen  
                     size=1,  
                     alpha=0.5)  
# Show plot  
p
```



```
In [184]: # Select (add) points to chart:  
p <- p + geom_point(aes(color = Source),  
                     size=2,  
                     shape=20,  
                     alpha=0.5)  
# Show plot  
p
```

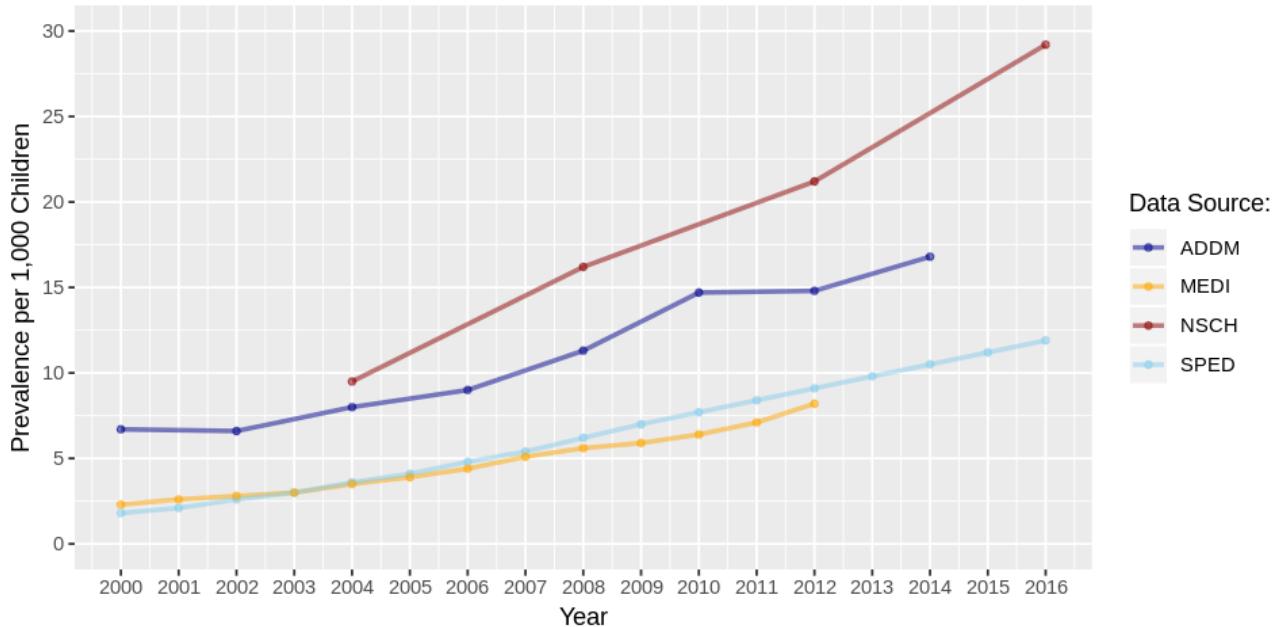


```
In [185]: # Customize line color and legend name:  
p <- p + scale_color_manual("Data Source:",  
                            labels = c('ADDM', 'MEDI', 'NSCH', 'SPED'),  
                            values = c("addm" = "darkblue",  
                                      "medi" = "orange",  
                                      "nsch" = "darkred",  
                                      "sped" = "skyblue"))  
# Show plot  
p
```

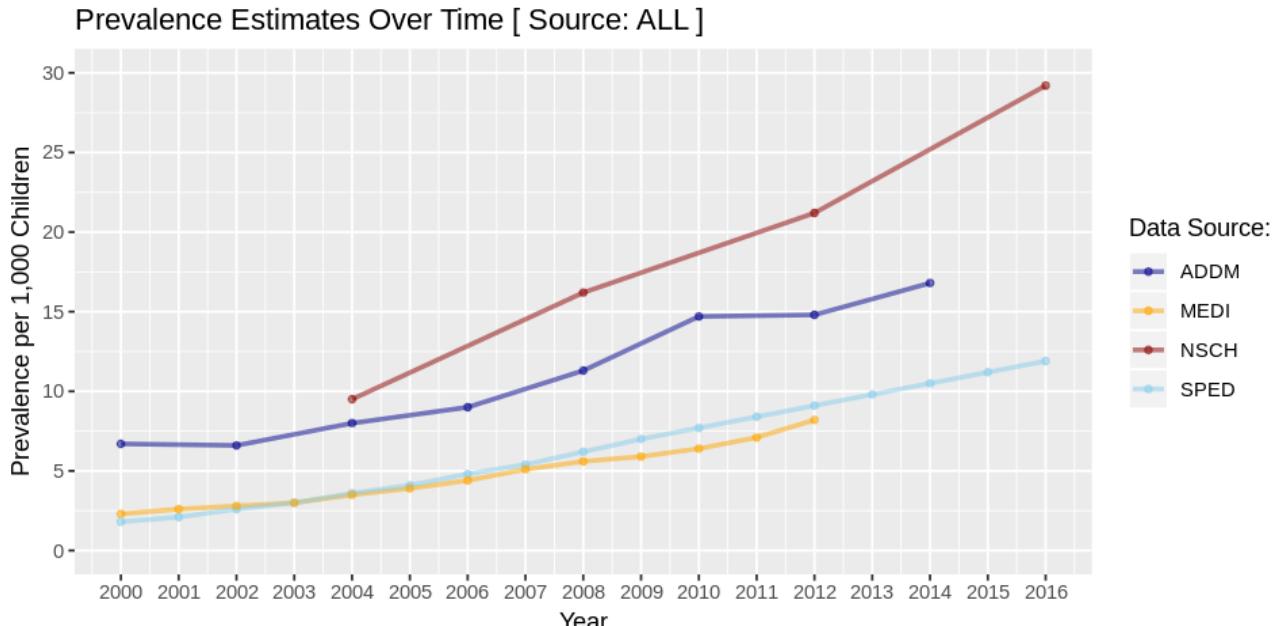


```
In [186]: # Adjust x and y axis, scale, limit and labels:
p <- p + scale_y_continuous(name = "Prevalence per 1,000 Children",
                             breaks = seq(0, 30, 5),
                             limits=c(0, 30)) +
  scale_x_continuous(name = "Year",
                     breaks = seq(2000, 2016, 1),
                     limits = c(2000, 2016))

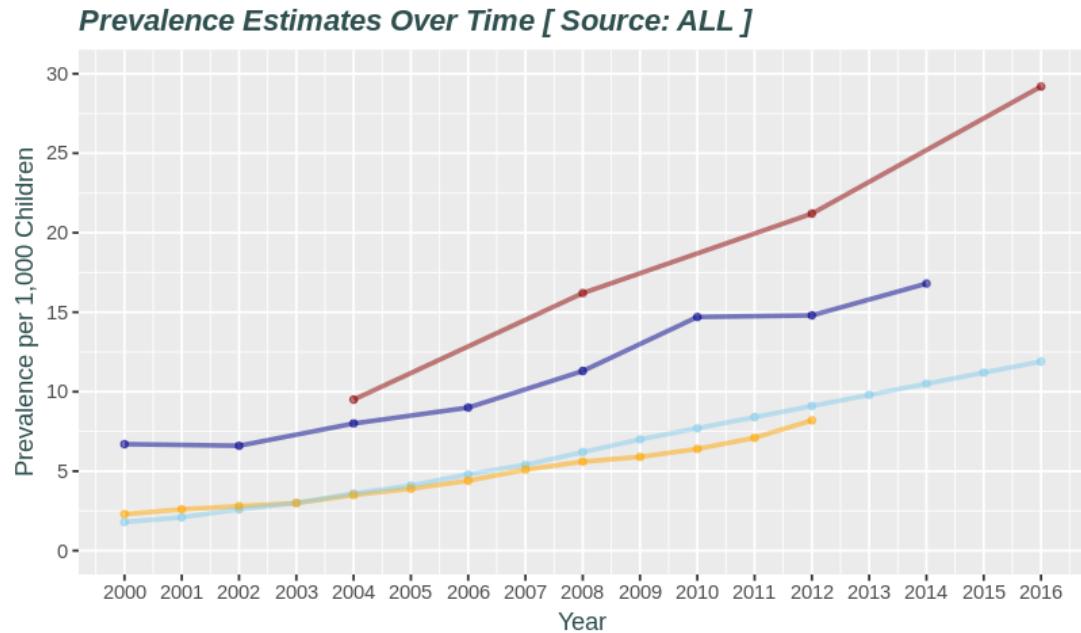
# Show plot
p
```



```
In [187]: # Customise chart title:
p <- p + ggtitle("Prevalence Estimates Over Time [ Source: ALL ]")
# Show plot
p
```



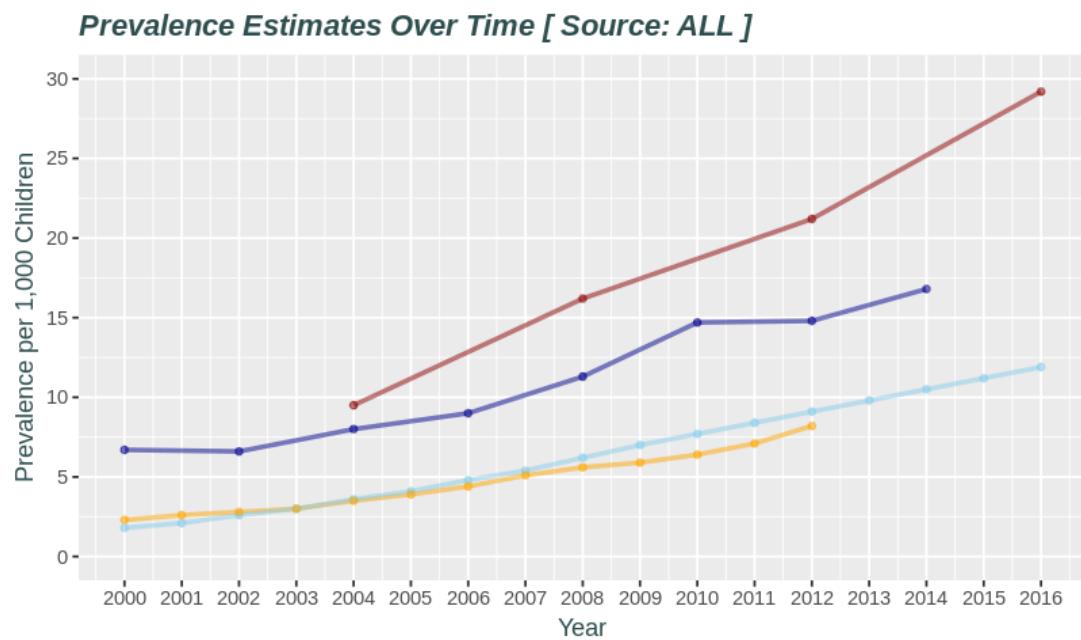
```
In [188]: # Customise chart title and axis labels:  
p <- p + theme(title = element_text(face = 'bold.italic', color = "darkslategray4",  
                 axis.title = element_text(face = 'plain', color = "darkslategray4"),  
                 axis.line = element_line(color = "darkslategray4"),  
                 panel.border = element_rect(colour = "darkslategray4", fill = "white"),  
                 panel.grid.major = element_line(colour = "darkslategray4"),  
                 panel.grid.minor = element_line(colour = "darkslategray4"),  
                 text = element_text(size = 12),  
                 plot.title = element_text(hjust = 0.5, size = 14),  
                 plot.subtitle = element_text(hjust = 0.5, size = 12),  
                 plot.background = element_rect(fill = "white"))  
# Show plot  
p
```



Consolidate above code into one chunk:

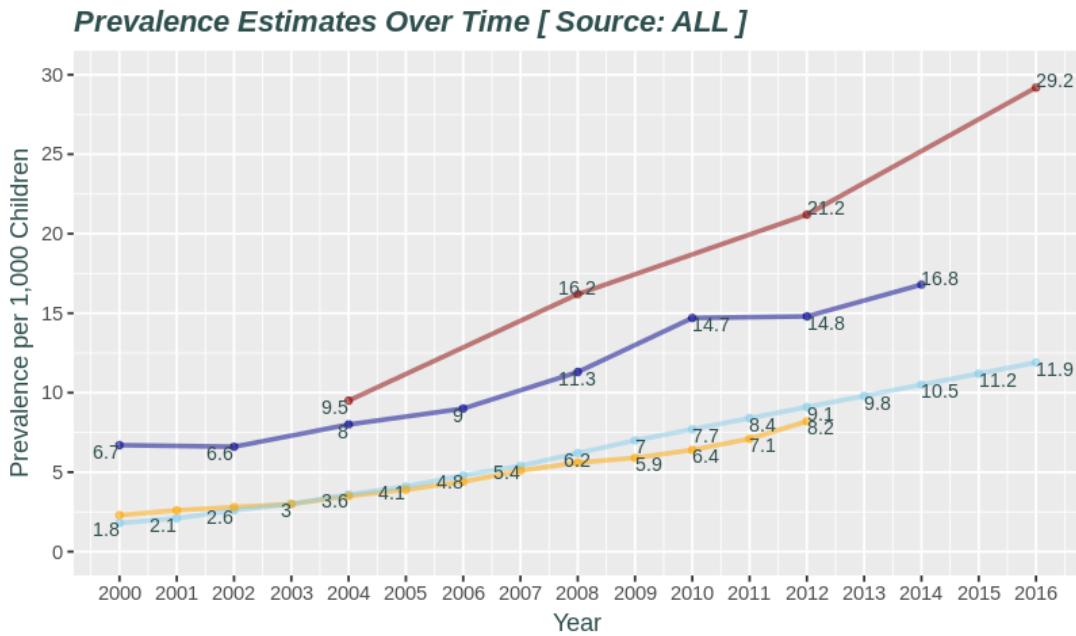
In [189]:

```
# -----
# Consolidate above code into one chunk
# -----
p <- ggplot(ASD_National, aes(x = Year, y = Prevalence)) +
  geom_line(aes(color = Source),
            linetype = "solid", # http://sape.inf.usi.ch/quick-reference/ggplot2/geom_line.html
            size=1,
            alpha=0.5) +
  geom_point(aes(color = Source),
             size=2,
             shape=20,
             alpha=0.5) +
  scale_color_manual("Data Source:",
                     labels = c('ADDM', 'MEDI', 'NSCH', 'SPED'),
                     values = c("addm" = "darkblue",
                               "medi" = "orange",
                               "nsch" = "darkred",
                               "sped" = "skyblue")) +
  scale_y_continuous(name = "Prevalence per 1,000 Children",
                     breaks = seq(0, 30, 5),
                     limits=c(0, 30)) +
  scale_x_continuous(name = "Year",
                     breaks = seq(2000, 2016, 1),
                     limits = c(2000, 2016)) +
  ggtile("Prevalence Estimates Over Time [ Source: ALL ]") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"))
# Show plot
p
```



Optionally, display data values/labels:

```
In [190]: # Optionally, display data values/labels
p + geom_text(aes(label = round(Prevalence, 1)), # Values are rounded for display
              vjust = "outward",
              #           nudge_y = 0.2, # optionally life the text
              hjust = "outward",
              check_overlap = TRUE,
              size = 3, # size of textual data label
              col = 'darkslategrey')
```



Data Visualisation (Enhanced) - Dynamic Visualisation with plotly

```
In [191]: if(!require(plotly)){install.packages("plotly")}
library(plotly)
```

Loading required package: plotly

Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

last_plot

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

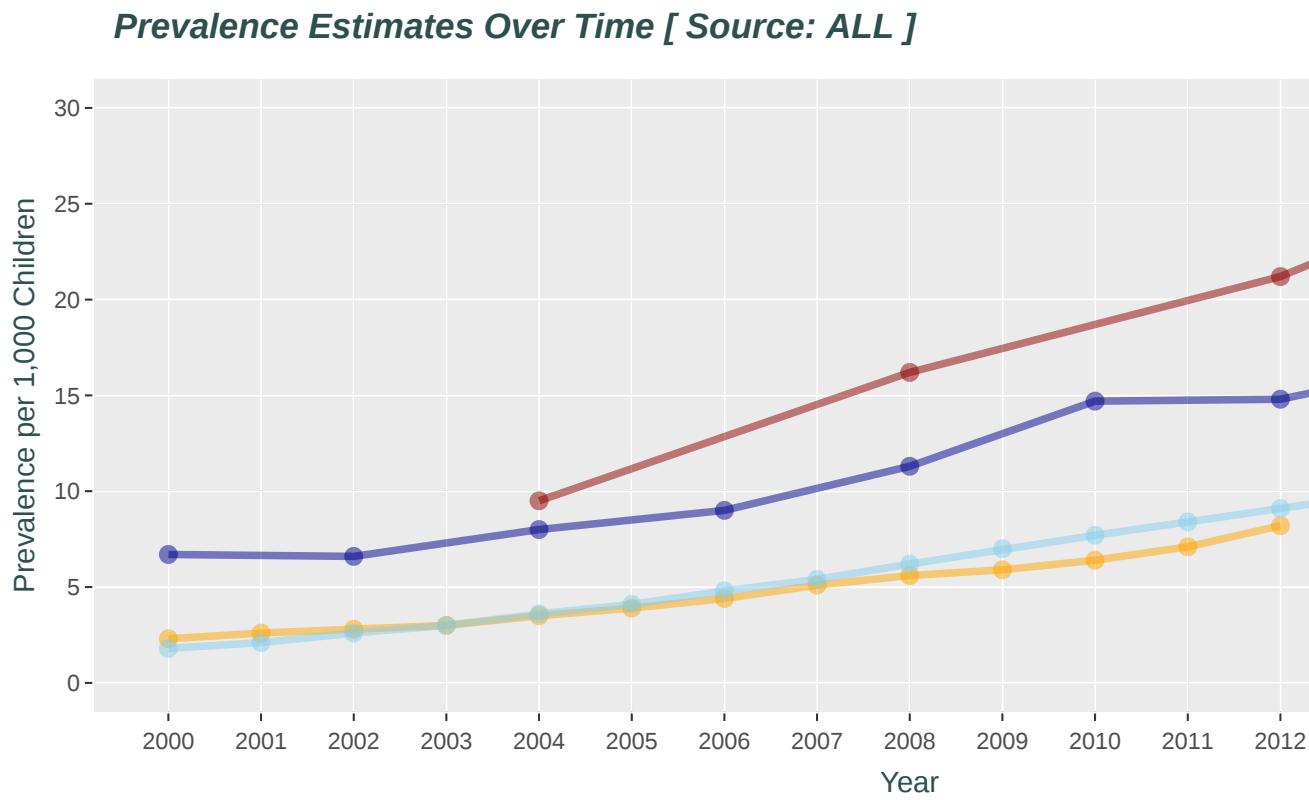
filter

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

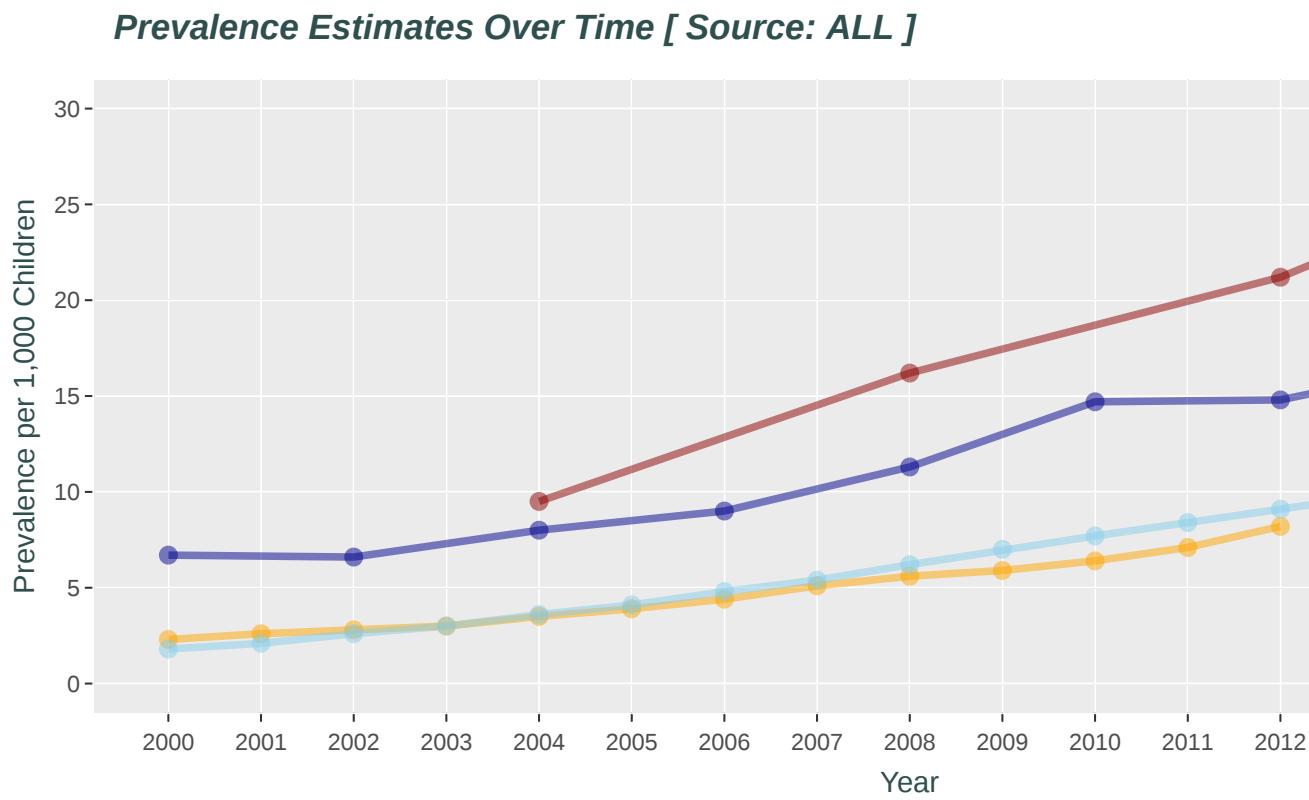
layout

Create ployly graph object from ggplot graph object:

```
In [192]: p_dynamic <- p  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```



```
In [193]: p_dynamic
```



Data Visualisation (Enhanced) - Use themes as aesthetic template

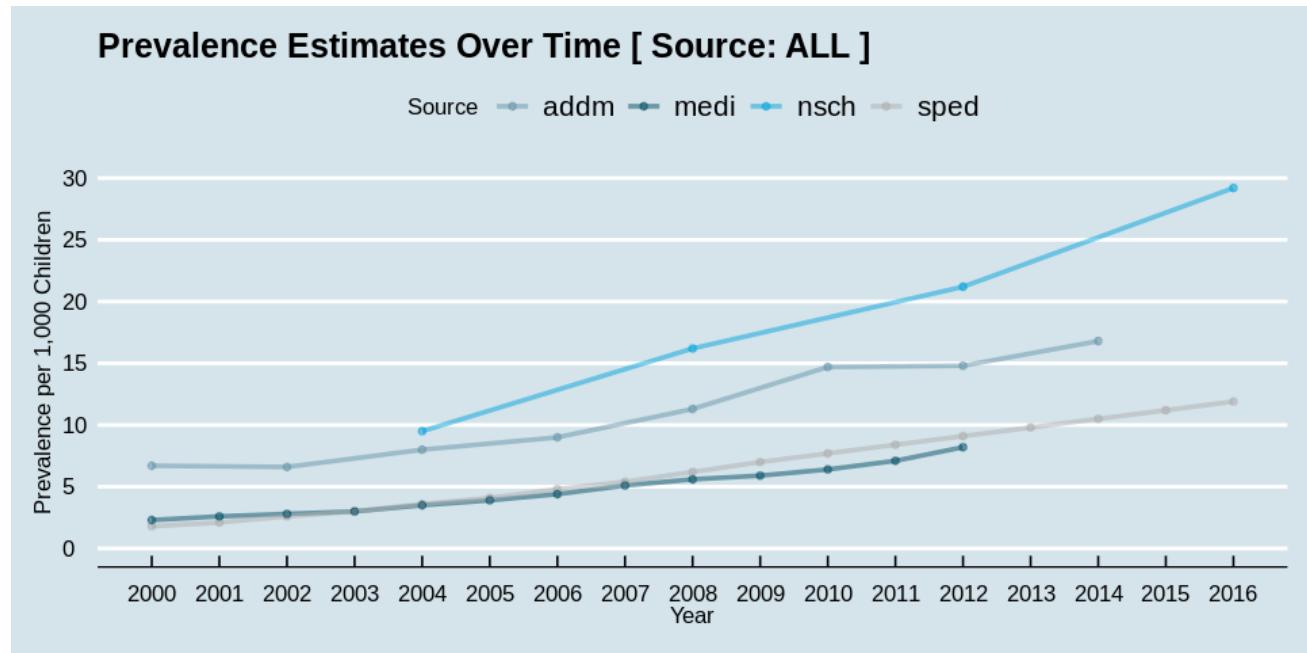
```
In [194]: if(!require(ggthemes)){install.packages("ggthemes")}  
library('ggthemes')
```

Loading required package: ggthemes

Theme of the Economist magazine:

```
In [195]: # Theme of the economist magazine:  
p + theme_economist() + scale_colour_economist()
```

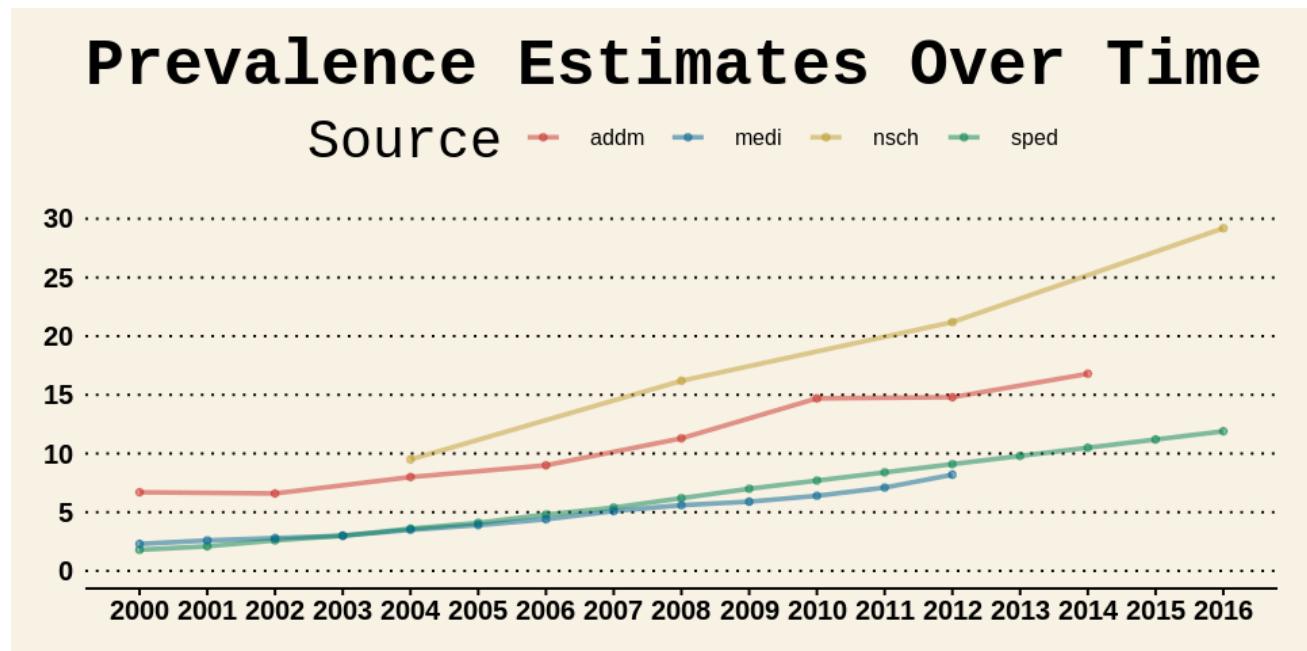
Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



Theme of the Wall Street Journal:

```
In [196]: # Theme of the Wall Street Journal:  
p + theme_wsj() + scale_colour_wsj("colors6")
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.

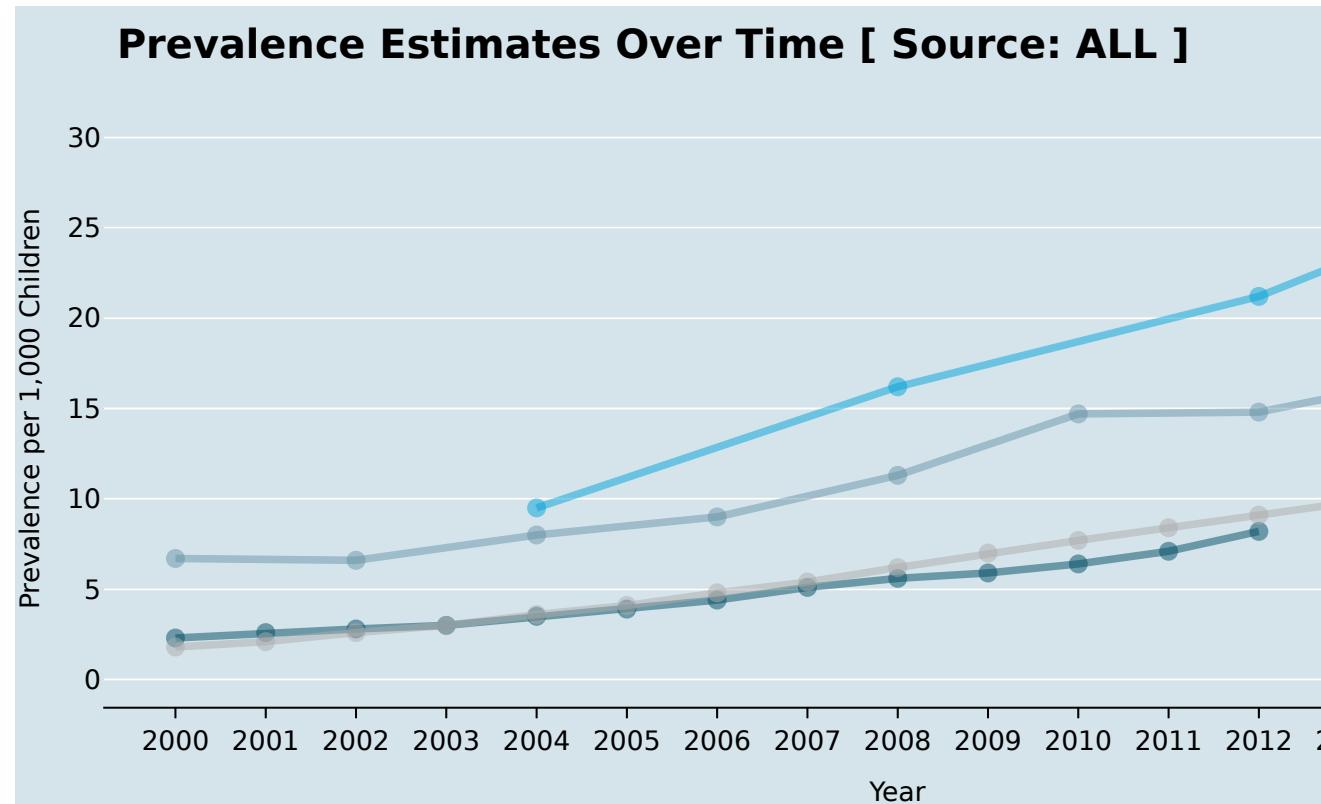


Dynamic chart with theme of the economist magazine:

In [197]: # Dynamic chart with theme of the economist magazine:

```
p_dynamic <- p + theme_economist() + scale_colour_economist()  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



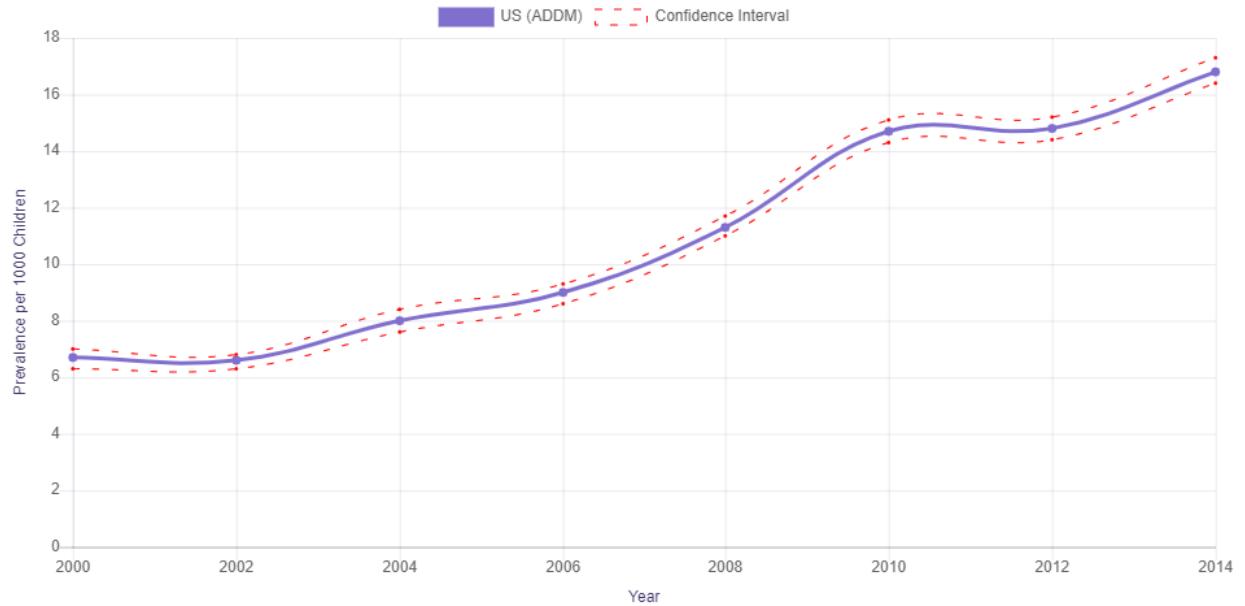
Data Visualisation (Enhanced) - [CDC] ADDM Network estimates for overall ASD prevalence in US over time [Source: ADDM] over [Year]

2014 ADDM NETWORK DATA

In this section, explore the most recent ADDM data, both overall and among certain demographic groups by study area.

ADDM Network estimates for overall ASD prevalence in US over time

with confidence interval



*ADDM data do not represent the entire state, only a selection of sites within the state.

**ADDM estimate = the total for all sites combined.

[†]NSCH data are not comparable over time as data collection methods changed and the data are not provided here. See technical notes for further details.

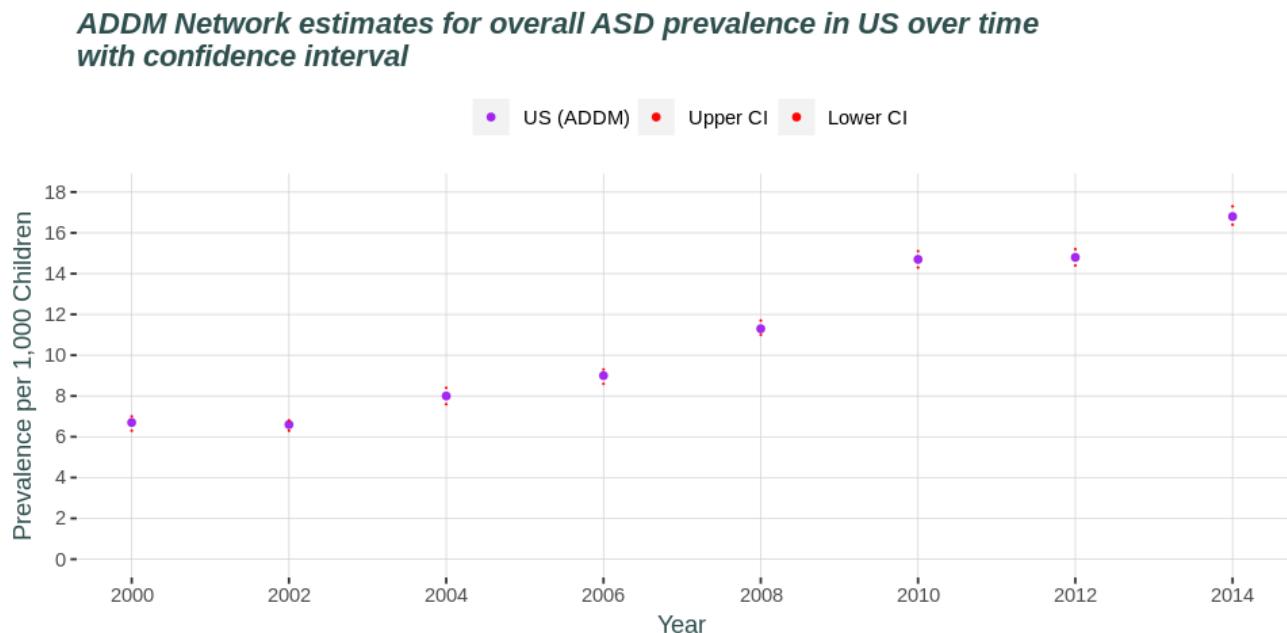
Data Visualisation (Enhanced) - [R] ADDM Network estimates for overall ASD prevalence in US over time [Source: ADDM] over [Year]

```
In [198]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

```
In [199]: # Filter only data of ADDM  
ASD_National_ADDM <- subset(ASD_National, Source == 'addm')
```

In [200]:

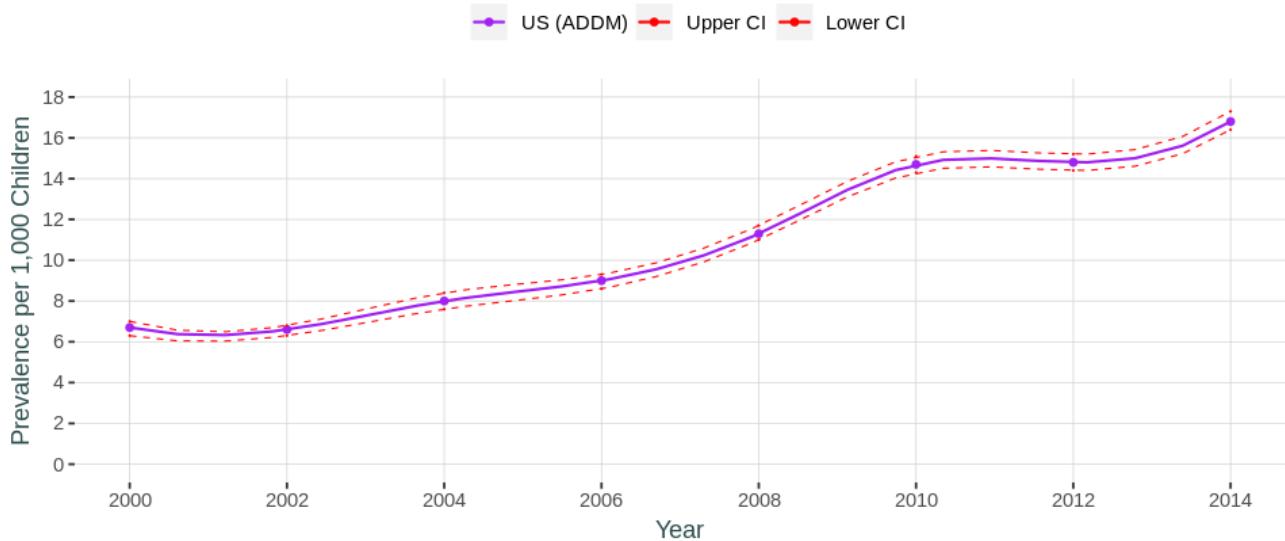
```
# -----  
# [addm] ADDM Network estimates for overall ASD prevalence in US over time  
# -----  
  
# Color:  
# 'ADDM_Average' "purple"  
  
p <- ggplot(ASD_National_ADDM, aes(x = Year, y = Prevalence)) +  
  geom_point(aes(y = Prevalence, color = 'ADDM_Average'), # Name for manual co  
              size=2,  
              shape=20,  
              alpha=0.95) +  
  # Add point for Upper.CI  
  geom_point(aes(y = Upper.CI, color = 'ADDM_U_CI'), # Name for manual colour  
              size=0.1,  
              shape=20,  
              alpha=0.95) +  
  # Add point for Lower.CI  
  geom_point(aes(y = Lower.CI, color = 'ADDM_L_CI'), # Name for manual colour  
              size=0.1,  
              shape=20,  
              alpha=0.95) +  
  scale_colour_manual(name="",  
                      labels = c("US (ADDM)", "Upper CI", "Lower CI"), # Names  
                      values = c(ADDM_Average="purple", ADDM_U_CI="red", ADDM_L_CI="blue"))  
# Add title, axis label, and axis scale  
p <- p + scale_y_continuous(name = "Prevalence per 1,000 Children",  
                            breaks = seq(0, 18, 2),  
                            limits=c(0, 18)) +  
  scale_x_continuous(name = "Year",  
                     breaks = seq(2000, 2014, 2),  
                     limits = c(2000, 2014)) +  
  ggtile("ADDM Network estimates for overall ASD prevalence in US over time\nwith confidence interval",  
        theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),  
              axis.title = element_text(face = 'plain', color = "darkslategrey"),  
              panel.background = element_blank(), # Remove chart background colour  
              legend.position = 'top',  
              panel.grid.major = element_line(size = 0.2, linetype = 'solid', colour = "black"))  
# Show plot  
p
```



```
In [201]: # Add smooth curve to go through date points, using interpolation with splines
# https://stackoverflow.com/questions/35205795/plotting-smooth-line-through-all-points-in-a-dataframe
spline_ADDM_Prevalence <- as.data.frame(spline(ASD_National_ADDM$Year, ASD_National_ADDM$Prevalence))
spline_ADDM_Prevalence_U_CI <- as.data.frame(spline(ASD_National_ADDM$Year, ASD_National_ADDM$Prevalence_U_CI))
spline_ADDM_Prevalence_L_CI <- as.data.frame(spline(ASD_National_ADDM$Year, ASD_National_ADDM$Prevalence_L_CI))

# Show plot
p + geom_line(data = spline_ADDM_Prevalence, aes(x = x, y = y, color = 'ADDM_Avg')) +
  geom_line(data = spline_ADDM_Prevalence_U_CI, aes(x = x, y = y, color = 'ADDM_U_CI')) +
  geom_line(data = spline_ADDM_Prevalence_L_CI, aes(x = x, y = y, color = 'ADDM_L_CI'))
```

ADDM Network estimates for overall ASD prevalence in US over time with confidence interval



Data Visualisation (Enhanced) - [R] REPORTED PREVALENCE VARIES BY SEX [Source: ADDM] over [Year]

```
In [202]: # Adjust in-line plot size to M x N
# options(repr.plot.width=8, repr.plot.height=4)
```

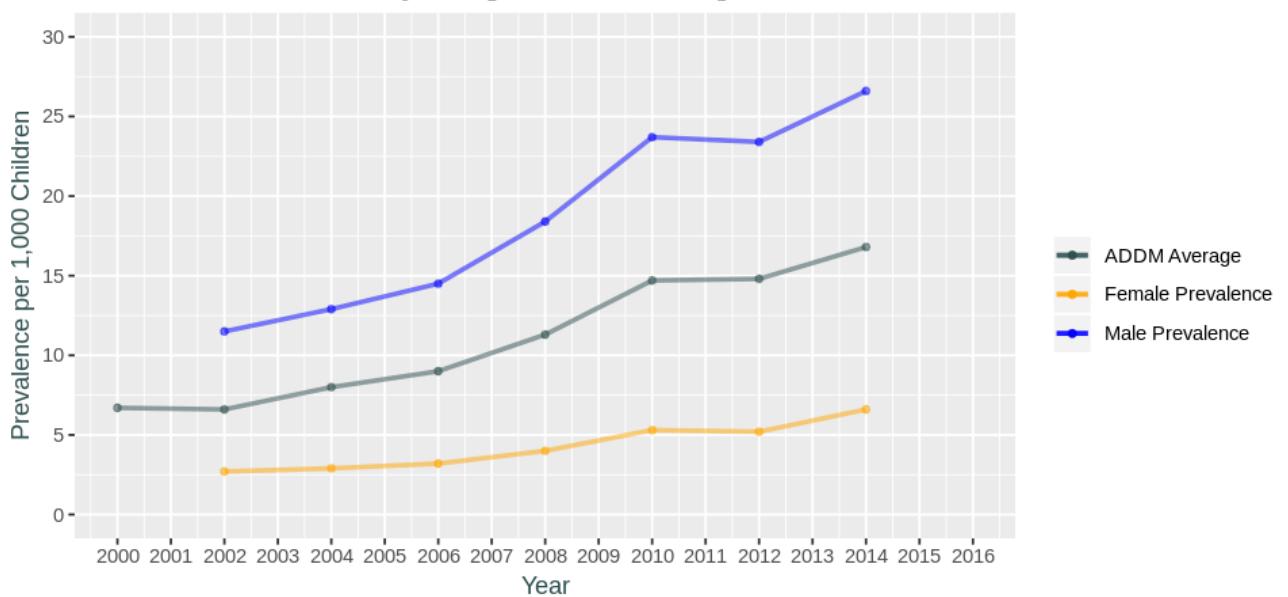
In [203]:

```
# -----
# [addm] < Prevalence Varies by Sex >
# -----  
  
# Color:  
# 'ADDM_Average' "darkslategrey"  
# 'Female_Prevalence' "orange"  
# 'Male_Prevalence' "blue"  
  
p <- ggplot(ASD_National_ADDM, aes(x = Year, y = Prevalence)) +  
  geom_line(aes(y = Prevalence, colour = 'ADDM_Average'),  
            linetype = "solid", # http://sape.inf.usi.ch/quick-reference/ggplot2/geom\_line.html,  
            size=1,  
            alpha=0.5) +  
  geom_point(aes(y = Prevalence, color = 'ADDM_Average'),  
             size=2,  
             shape=20,  
             alpha=0.5) +  
  # Add line for Female  
  geom_line(aes(y = Female.Prevalence, colour = 'Female_Prevalence'),  
            linetype = "solid", # http://sape.inf.usi.ch/quick-reference/ggplot2/geom\_line.html,  
            size=1,  
            alpha=0.5) +  
  geom_point(aes(y = Female.Prevalence, color = 'Female_Prevalence'),  
             size=2,  
             shape=20,  
             alpha=0.5) +  
  # Add line for Male  
  geom_line(aes(y = Male.Prevalence, colour = 'Male_Prevalence'),  
            linetype = "solid", # http://sape.inf.usi.ch/quick-reference/ggplot2/geom\_line.html,  
            size=1,  
            alpha=0.5) +  
  geom_point(aes(y = Male.Prevalence, color = 'Male_Prevalence'),  
             size=2,  
             shape=20,  
             alpha=0.5) +  
  scale_colour_manual(name="",  
                      labels = c("ADDM Average", "Female Prevalence", "Male Prevalence"),  
                      values = c(ADDM_Average="darkslategrey", Female_Prevalence="orange", Male_Prevalence="blue"))  
# Add title, axis label, and axis scale  
p <- p + scale_y_continuous(name = "Prevalence per 1,000 Children",  
                             breaks = seq(0, 30, 5),  
                             limits=c(0, 30)) +  
  scale_x_continuous(name = "Year",  
                     breaks = seq(2000, 2016, 1),  
                     limits = c(2000, 2016)) +  
  ggtitle("Prevalence Estimates by Sex [ Source: ADDM ]") +  
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),  
        axis.title = element_text(face = 'plain', color = "darkslategrey"))  
# Show plot  
p
```

Warning message:

"Removed 1 rows containing missing values (geom_path)."Warning message:
"Removed 1 rows containing missing values (geom_point)."Warning message:
"Removed 1 rows containing missing values (geom_path)."Warning message:
"Removed 1 rows containing missing values (geom_point)."

Prevalence Estimates by Sex [Source: ADDM]



In [204]: `# Apply theme`

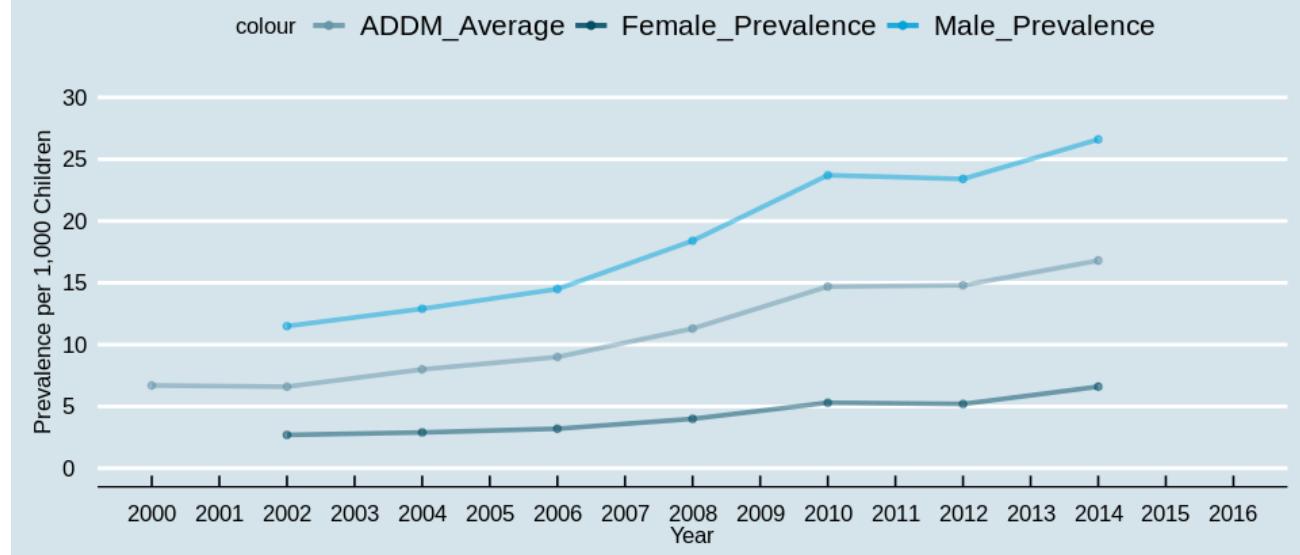
```
p + theme_economist() + scale_colour_economist() # p + theme_wsj() + scale_col
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.

Warning message:

```
"Removed 1 rows containing missing values (geom_path)."Warning message:  
"Removed 1 rows containing missing values (geom_point)."Warning message:  
"Removed 1 rows containing missing values (geom_path)."Warning message:  
"Removed 1 rows containing missing values (geom_point)."
```

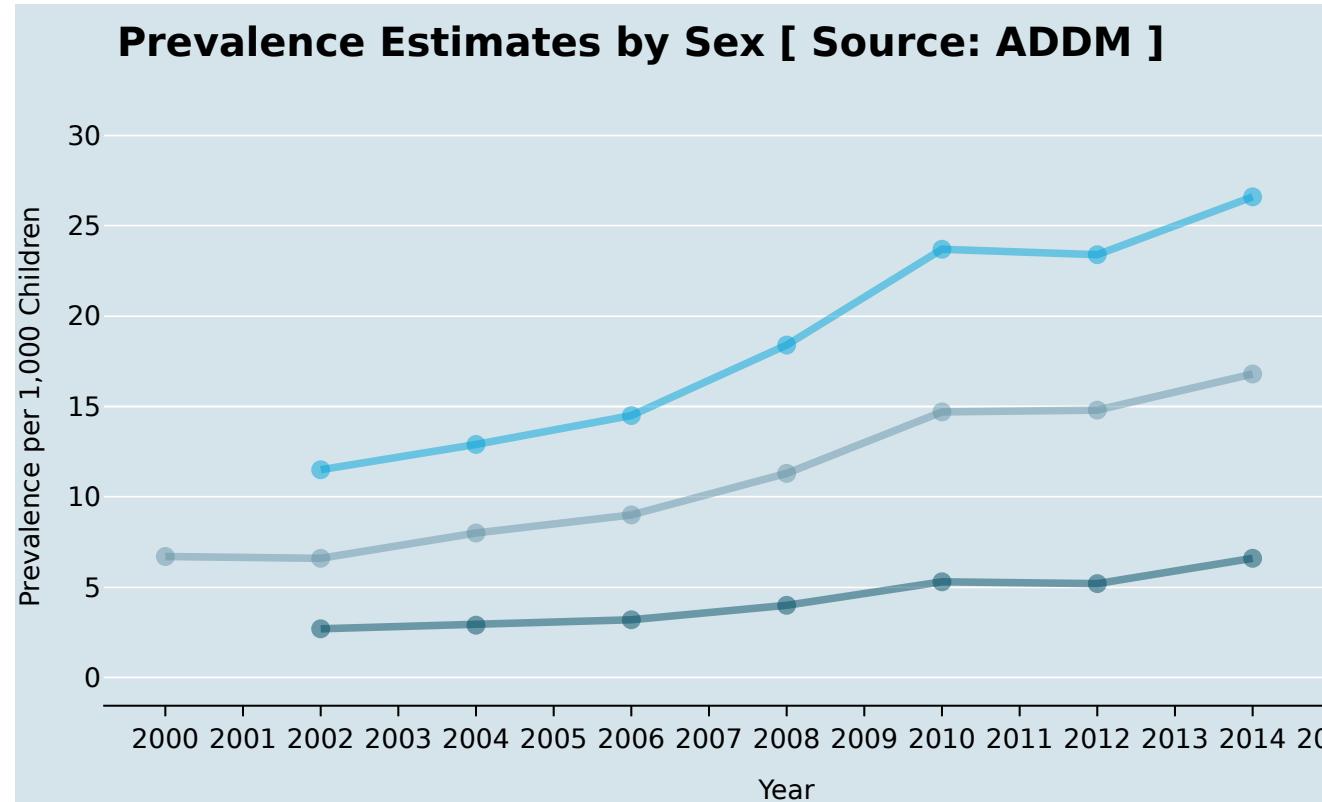
Prevalence Estimates by Sex [Source: ADDM]



In [205]: # Dynamic chart:

```
p_dynamic <- p + theme_economist() + scale_colour_economist()  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



Quiz:

Add 95% Confidence Interval to above plot (Use ggplot)

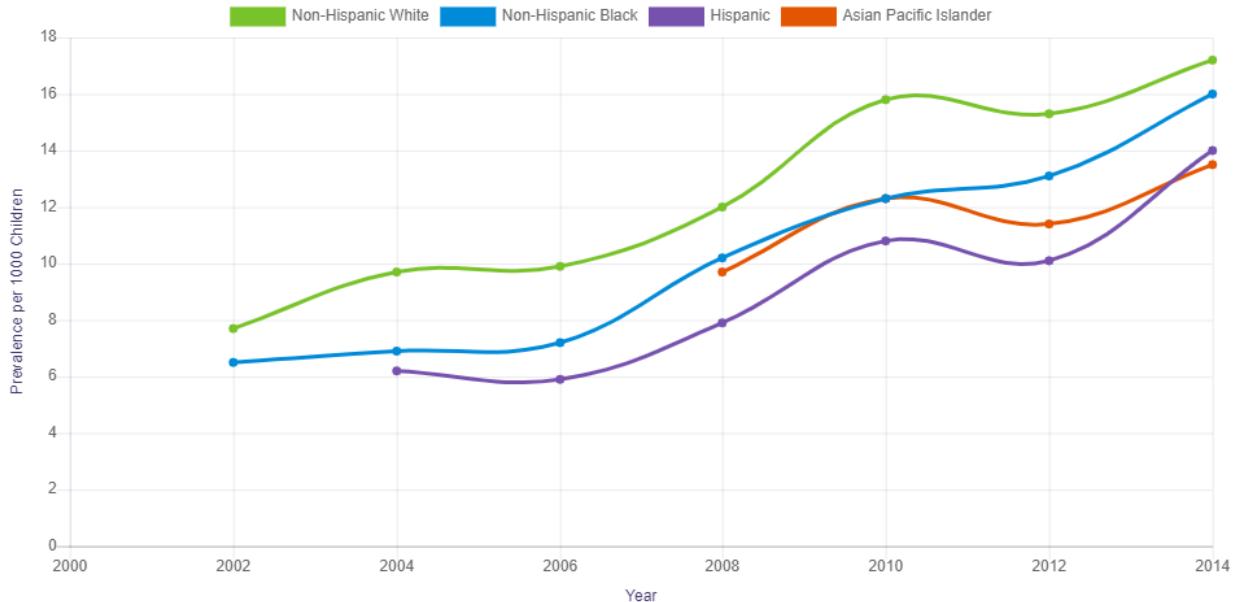
In [206]: # Write your code below and press Shift+Enter to execute

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

Data Visualisation (Enhanced) - [CDC] REPORTED PREVALENCE VARIES BY RACE AND ETHNICITY

Prevalence Estimates by Race/Ethnicity

Show ADDM prevalence estimates* by race/ethnicity for: U.S. or Total+ ▾



Note: Click the icons and racial/ethnic groups above the chart to hide or unhide data. Hover your mouse over data points to show prevalence by year.

*ADDM data do not represent the entire state, only a selection of sites within the state.

+ADDM estimate = the total for all sites combined.

Data Visualisation (Enhanced) - [R] REPORTED PREVALENCE VARIES BY RACE AND ETHNICITY [Source: ADDM] With Average

```
In [207]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

In [208]:

```

Non_Hispanic_Black ="deepskyblue3",
Non_Hispanic_White ="chartreuse3"))

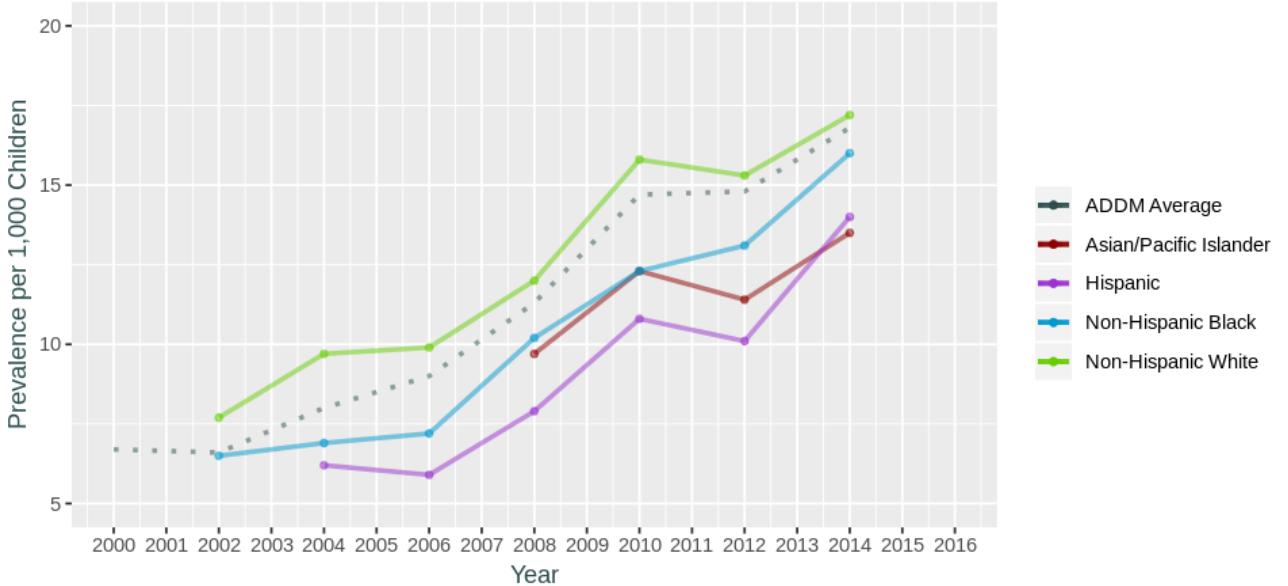
# Add title, axis label, and axis scale
p <- p + scale_y_continuous(name = "Prevalence per 1,000 Children",
                             breaks = seq(5, 20, 5),
                             limits=c(5, 20)) +
  scale_x_continuous(name = "Year",
                     breaks = seq(2000, 2016, 1),
                     limits = c(2000, 2016)) +
  ggtitle("Prevalence Estimates by Race/Ethnicity [ Source: ADDM ]") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"))

# Show plot
p

```

Warning message:
 "Removed 4 rows containing missing values (geom_path)."Warning message:
 "Removed 4 rows containing missing values (geom_point)."Warning message:
 "Removed 2 rows containing missing values (geom_path)."Warning message:
 "Removed 2 rows containing missing values (geom_point)."Warning message:
 "Removed 1 rows containing missing values (geom_path)."Warning message:
 "Removed 1 rows containing missing values (geom_point)."Warning message:
 "Removed 1 rows containing missing values (geom_path)."Warning message:
 "Removed 1 rows containing missing values (geom_point)."Warning message:
 "Removed 1 rows containing missing values (geom_point)."

Prevalence Estimates by Race/Ethnicity [Source: ADDM]

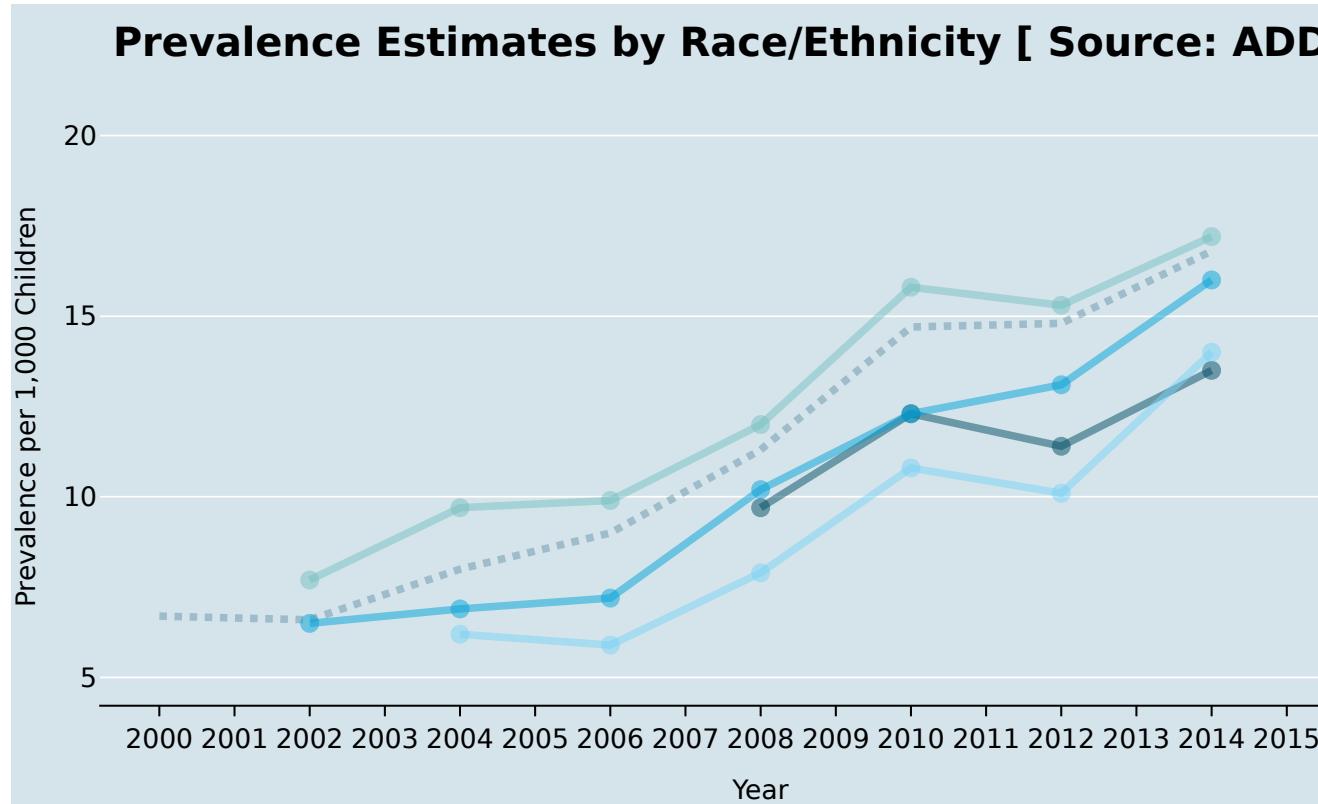


In [209]: # Apply theme
 # p + theme_economist() + scale_colour_economist() # p + theme_wsj() + scale_c

In [210]: # Dynamic chart:

```
p_dynamic <- p + theme_economist() + scale_colour_economist()  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



Quiz:

Change above zig-zag lines to spline/smooth lines.

Hints: Refer to ADDM Network estimates for overall ASD prevalence in US over time.

In [211]: # Write your code below and press Shift+Enter to execute

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

0

Data Visualisation (Enhanced) - US. State Level Data Processing

In [212]:

```
# -----  
# Dataset: US. State Level Children ASD Prevalence  
# -----  
  
ASD_State <- read.csv("../dataset/ADV_ASD_State.csv", stringsAsFactors = FALSE)  
  
# Obtain number of rows and number of columns/features/variables  
dim(ASD_State)  
# Obtain overview (data structure/types)  
str(ASD_State)
```

1692 48

Data Visualisation (Enhanced) - US. State Level Data Pre-Process data

Pre-Process data: Missing data

In [213]:

```
# Count missing values in dataframe:  
sum(is.na(ASD_State)) # No missing data recognised by R (NA)  
# Define several offending strings  
na_strings <- c("", "No data", "NA", "N A", "N / A", "N/A", "N/ A", "Not Available", "N/A.", ".N/A", "N/A .")  
# Replace these defined missing values to R's internal NA  
ASD_State = replace_with_na_all(ASD_State, condition = ~.x %in% na_strings)  
# Count missing values in dataframe:  
sum(is.na(ASD_State))
```

14454

28992

Remove invalid unicode char/string: \x92

In [214]:

```
# Remove invalid unicode char/string: \x92  
ASD_State$Source_Full1[ASD_State$Source_Full1 == "National Survey of Children's Health"] = ""
```

Delete/Drop variable by index: column from 14 to 26, 29, and 30

```
In [215]: cbind(names(ASD_State), c(1:length(names(ASD_State))))
```

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 [216]: # Delete/Drop variable by index: column from 14 to 26, 29, and 30  
# names(ASD_State)  
ASD_State <- ASD_State[ -c(14:26, 29, 30) ]
```

Create new variables

```
In [217]: # Create one new variable: Source_UC as uppercase of Source  
ASD_State$Source_UC <- toupper(ASD_State$Source)  
# Create one new variable: Source_Full3 by combining Source_UC and Source_Full1  
ASD_State$Source_Full3 <- paste(ASD_State$Source_UC, ASD_State$Source_Full1)
```

Create one new ordinal categorical variable: Prevalence_Rank2 ("Low", "High") by binning Prevalence

```
In [218]: # Recode Risk into category from Prevalence  
  
# Low [0, 5)  
# High [5, +oo)  
  
ASD_State$Prevalence_Risk2[ASD_State$Prevalence < 5] = "Low"  
ASD_State$Prevalence_Risk2[ASD_State$Prevalence >= 5] = "High"  
#  
# head(ASD_State)
```

Warning message:
"Unknown or uninitialized column: 'Prevalence_Risk2'."

Create one new ordinal categorical variable: Prevalence_Rank4 ("Low", "Medium", "High", "Very High") by binning Prevalence

```
In [219]: # Recode Risk into category from Prevalence
```

```
# Low [0, 5)
# Medium [5, 10)
# High [10, 20)
# Very High [20, +oo)

ASD_State$Prevalence_Risk4 = "Very High"
ASD_State$Prevalence_Risk4[ASD_State$Prevalence < 20] = "High"
ASD_State$Prevalence_Risk4[ASD_State$Prevalence < 10] = "Medium"
ASD_State$Prevalence_Risk4[ASD_State$Prevalence < 5] = "Low"
#
# head(ASD_State)
```

Convert to correct data types

```
In [220]: str(ASD_State)
```

```
Classes 'tbl_df', 'tbl' and 'data.frame': 1692 obs. of 37 variables:
 $ State : chr "AZ" "GA" "MD" "NJ" ...
 $ Denominator : int 45322 43593 21532 29714 24535
 23065 35472 45113 36472 11020 ...
 $ Prevalence : num 6.5 6.5 5.5 9.9 6.3 4.5 3.3
 6.2 6.9 5.9 ...
 $ Lower.CI : num 5.8 5.8 4.6 8.9 5.4 3.7 2.7
 5.5 6.1 4.6 ...
 $ Upper.CI : num 7.3 7.3 6.6 11.1 7.4 5.5 3.9
 7 7.8 7.5 ...
 $ Year : int 2000 2000 2000 2000 2000 2000
 2002 2002 2002 2002 ...
 $ Source : chr "addm" "addm" "addm" "addm"
 ...
 $ Source_Full1 : chr "Autism & Developmental Disabilities Monitoring Network" ...
 $ State_Full1 : chr "Arizona" "Georgia" "Marylan
" "New Jersey"
```

```
In [221]: # cbind(names(ASD_State), c(1:length(names(ASD_State))))
```

Convert variables to numeric

```
In [222]: # Convert Prevalence and CIs from categorical/chr to numeric
ix <- 13:33 # define an index
ASD_State[ix] <- lapply(ASD_State[ix], as.numeric)
```

Convert variables to categorical/factor

```
In [223]: # Convert Source from categorical/chr to categorical/factor
ix <- c(1, 7, 8, 9, 10, 34, 35) # define an index
ASD_State[ix] <- lapply(ASD_State[ix], as.factor)

# Create new ordered factor Year_Factor from Year
ASD_State$Year_Factor <- factor(ASD_State$Year, ordered = TRUE)
```

Convert Prevalence_Rank2 & Prevalence_Rank4 to ordered factor

```
In [224]: # Convert to factor  
ASD_State$Prevalence_Risk2 = factor(ASD_State$Prevalence_Risk2, ordered=TRUE,  
                                     levels=c("Low", "High"))  
  
# Convert to factor  
ASD_State$Prevalence_Risk4 = factor(ASD_State$Prevalence_Risk4, ordered=TRUE,  
                                     levels=c("Low", "Medium", "High", "
```

```
In [225]: # Display unique values (levels) of a factor categorical  
lapply(select_if(ASD_State, is.factor), levels)
```

\$State

```
'AK'  'AL'  'AR'  'AZ'  'CA'  'CO'  'CT'  'DC'  'DE'  'FL'  'GA'  'HI'  'IA'  'ID'  'IL'  'IN'  'KS'  
'KY'  'LA'  'MA'  'MD'  'ME'  'MI'  'MN'  'MO'  'MS'  'MT'  'NC'  'ND'  'NE'  'NH'  'NJ'  'NM'  
'NV'  'NY'  'OH'  'OK'  'OR'  'PA'  'RI'  'SC'  'SD'  'TN'  'TX'  'UT'  'VA'  'VT'  'WA'  'WI'  
'WV'  'WY'
```

\$Source

```
'addm'  'medi'  'nsch'  'sped'
```

\$Source_Full1

```
'Autism & Developmental Disabilities Monitoring Network'  'Medicaid'  
'National Survey of Children's Health'  'Special Education Child Count'
```

\$State_Full1

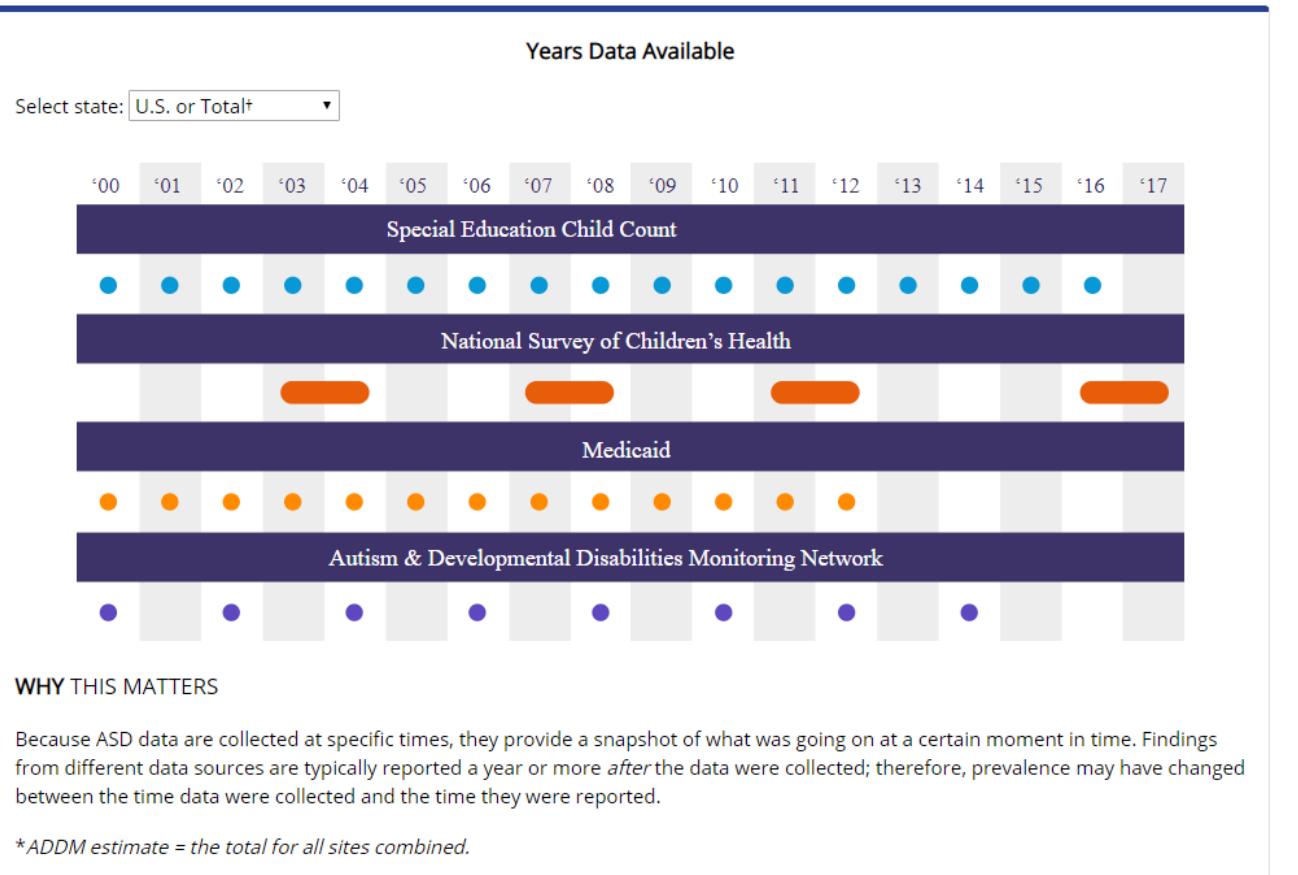
```
'Alabama'  'Alaska'  'Arizona'  'Arkansas'  'California'  'Colorado'  'Connecticut'  'Delaware'  
'District of Columbia'  'Florida'  'Georgia'  'Hawaii'  'Idaho'  'Illinois'  'Indiana'  'Iowa'  'Kansas'  
'Kentucky'  'Louisiana'  'Maine'  'Maryland'  'Massachusetts'  'Michigan'  'Minnesota'  
'Mississippi'  'Missouri'  'Montana'  'Nebraska'  'Nevada'  'New Hampshire'  'New Jersey'
```

Optionally, export the processed dataframe data to CSV file.

```
In [226]: write.csv(ASD_State, file = ".../dataset/ADV_ASD_State_R.csv", row.names = FALSE)
```

```
In [227]: # 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)
```

Data Visualisation (Enhanced) - US. State Level Data Visualisation



Above chart shows at data source level, we'd also like to know State level data availability. How?

Data Visualisation (Enhanced) - [R] Explore the Data [Years Data Available by State]

```
In [228]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=12)
```

```
In [229]: # -----
# [State] < Years Data Available by State >
# -----
p <- ggplot(ASD_State, aes(x = Source, fill = Source)) +
  geom_bar() + theme(axis.text.x=element_blank(), # Hide axis
                     axis.ticks.x=element_blank(), # Hide axis
                     axis.text.y=element_blank(), # Hide axis
                     axis.ticks.y=element_blank(), # Hide axis
                     panel.background = element_blank(), # Remove panel background
                     legend.position="top",
                     strip.text.y = element_text(angle=0) # Rotate text to horizontal
  ) +
  scale_fill_manual("Data Source:", values = c("addm" = "darkblue",
                                              "medi" = "orange",
                                              "nsch" = "darkred",
                                              "sped" = "skyblue")) +
  facet_grid(facets = State_Full2 ~ Year) +
  labs(x="", y="", title="Years Data Available by State") # layers of graphics
```

```
In [230]: # Below plot may run for a while  
# Show plot  
p
```

Years Data Available by State



Filter and create dataframe of different data sources, for easy data access

```
In [231]: # Filter and create dataframe of different data sources, for easy data access  
ASD_State_ADDM <- subset(ASD_State, Source == 'addm')  
ASD_State_MEDI <- subset(ASD_State, Source == 'medi')  
ASD_State_NSCH <- subset(ASD_State, Source == 'nsch')  
ASD_State_SPED <- subset(ASD_State, Source == 'sped')
```

Data Visualisation (Enhanced) - [R] Explore the Data Years Data Available by State [Source: ADDM]

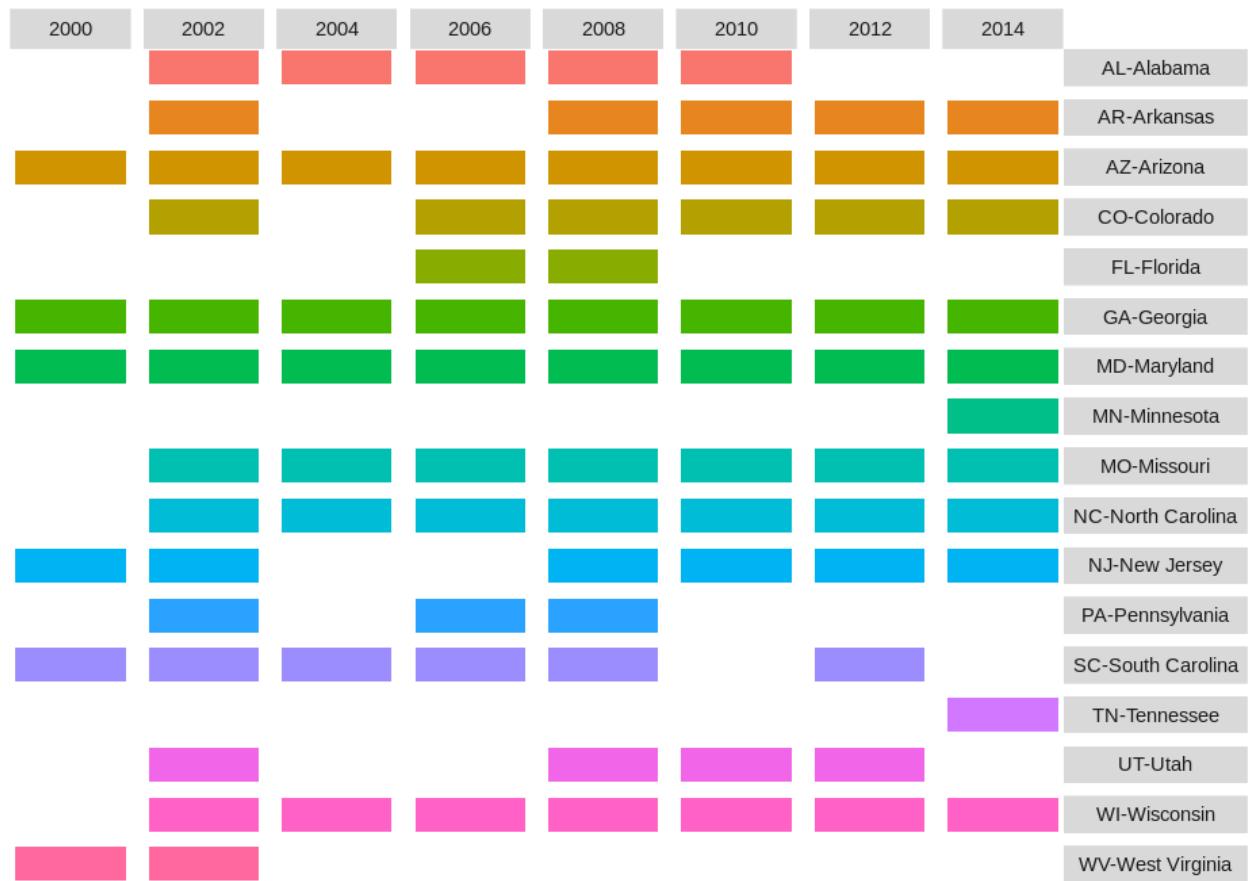
```
In [232]: # Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=6)
```

Years Data Available by State [Source: ADDM]

```
In [233]: # Years Data Available by State [ Source: ADDM ]  
p <- ggplot(ASD_State_ADDM, aes(x = 1, fill = State_Full2)) +  
  geom_bar() + theme(axis.text.x=element_blank(), # Hide axis  
                     axis.ticks.x=element_blank(), # Hide axis  
                     axis.text.y=element_blank(), # Hide axis  
                     axis.ticks.y=element_blank(), # Hide axis  
                     panel.background = element_blank(), # Remove panel background  
                     legend.position="none",  
                     strip.text.y = element_text(angle=0) # Rotate text to horizontal  
  ) +  
  facet_grid(facets = State_Full2 ~ Year_Factor) +  
  labs(x="", y="", title="Years Data Available by State [ Source: ADDM ]") # Labels
```

```
In [234]: # Show plot  
p
```

Years Data Available by State [Source: ADDM]



Quiz:

Create Years Data Available by State [Source: XXXX] for other three data sources:

```
In [235]: # Write your code below and press Shift+Enter to execute
```

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

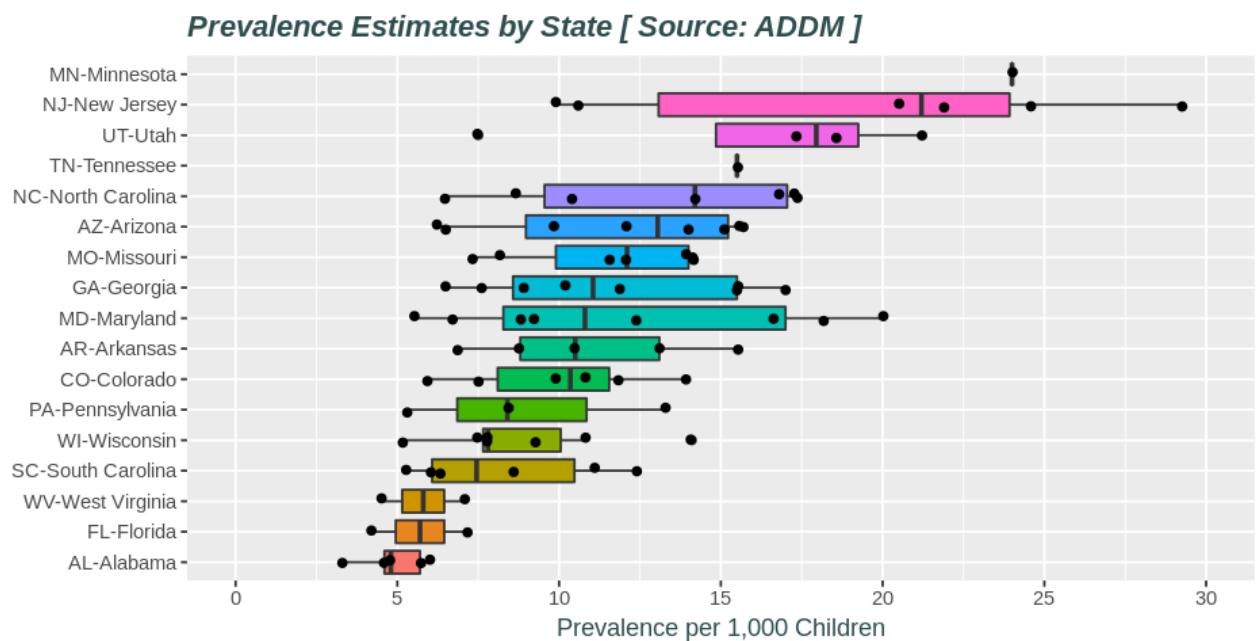
Data Visualisation (Enhanced) - [R] REPORTED PREVALENCE VARIES BY GEOGRAPHIC LOCATION (States) Prevalence Estimates by State [Source: ADDM]

```
In [236]: # Adjust in-line plot size to M x N  
options(repr.plot.width=8, repr.plot.height=4)
```

Visualise: **Prevalence Estimates by State [Source: ADDM]**

```
In [237]: # Prevalence Estimates by State [ Source: ADDM ] , aggregated for different years
p <- ggplot(ASD_State_ADDM, aes(x = reorder(State_Full2, Prevalence, FUN = median),
                                    y = Prevalence)) +
  geom_boxplot(aes(fill = reorder(State_Full2, Prevalence, FUN = median))) +
  scale_fill_discrete(guide = guide_legend(title = "US. States")) + # Legend
# geom_boxplot(fill = 'darkslategrey', alpha = 0.2) +
  scale_y_continuous(name = "Prevalence per 1,000 Children",
                     breaks = seq(0, 30, 5),
                     limits=c(0, 30)) +
  scale_x_discrete(name = "") +
  ggttitle("Prevalence Estimates by State [ Source: ADDM ]") +
  theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
        axis.title = element_text(face = 'plain', color = "darkslategrey"),
        legend.position = 'none') +
  coord_flip() + # Rotate chart
  geom_jitter(position=position_jitter(0.1)) # Add actual data points
```

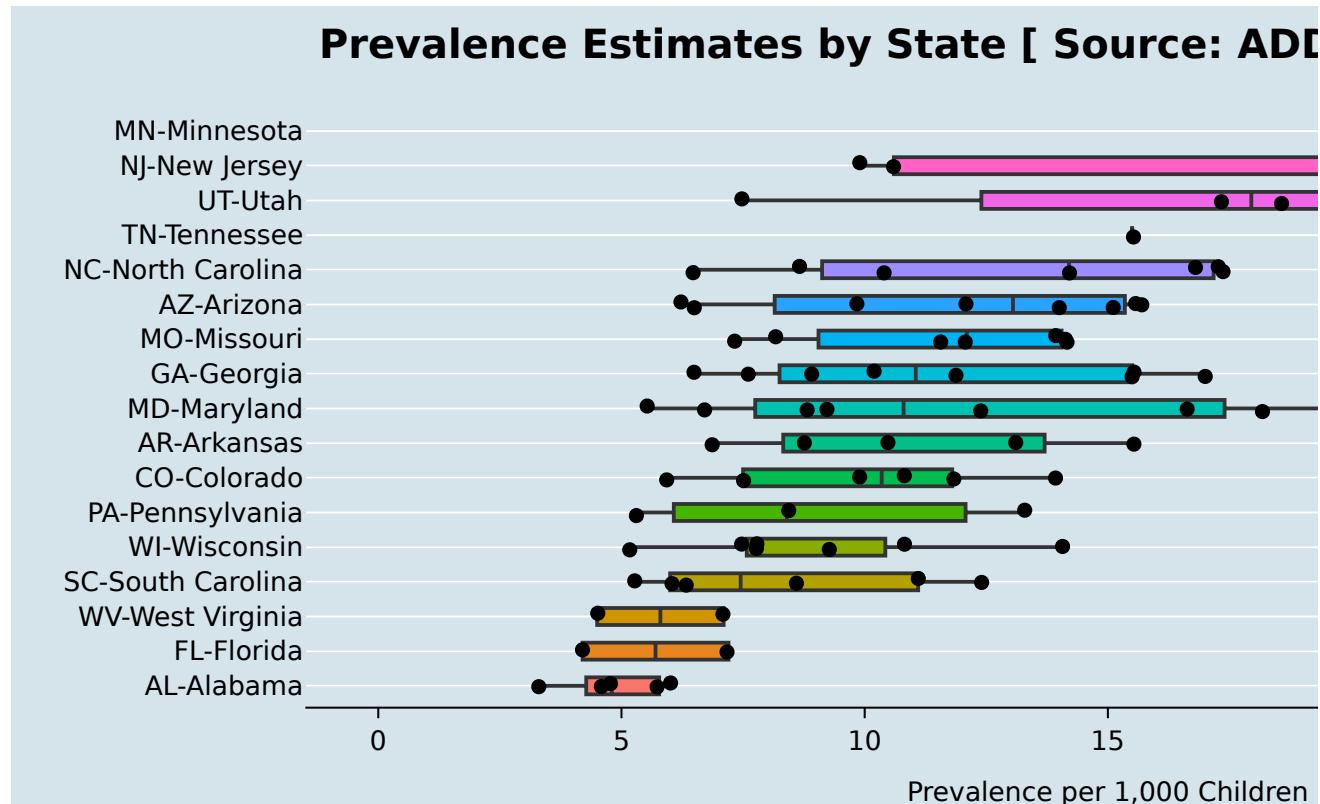
```
In [238]: # Show plot
p
```



```
In [239]: # Theme of the economist magazine:
# p + theme_economist() + scale_colour_economist() + theme(legend.position = 'top')
```

In [240]: # Dynamic chart

```
p_dynamic <- p + theme_economist() + scale_colour_economist() + theme(legend.position = "none")
p_dynamic <- ggplotly(p_dynamic)
p_dynamic
```



Quiz:

Create Prevalence Estimates by State [Source: XXXX] for other three data sources:

In [241]: # Write your code below and press Shift+Enter to execute

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

Data Visualisation (Enhanced) - [R] US. State Level No. Children Surveyed by State [Source: ADDM] [Year 2014]

In [242]: # Adjust in-line plot size to M x N

```
options(repr.plot.width=8, repr.plot.height=4)
```

Visualise: No. Children Surveyed by State [Source: ADDM] [Year 2014]

In [243]: # All State Prevalence data with: Source == 'addm' & Year == 2014

```
# filter using dataframe: ASD_State_ADDM
```

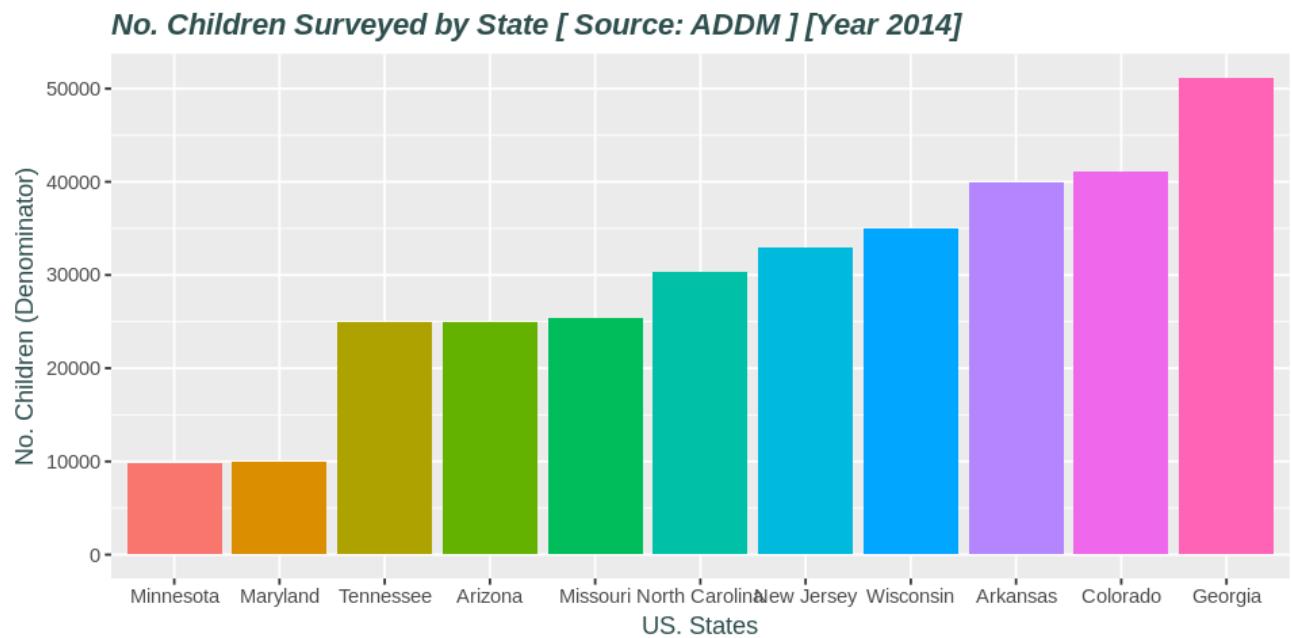
```
ASD_State_Subset <- subset(ASD_State_ADDM, Year == 2014)
```

```
# or filter using dataframe: ASD_State
```

```
ASD_State_Subset <- subset(ASD_State, Source == 'addm' & Year == 2014)
```

```
In [244]: # Bar plot/chart for < No. Children surveyed by State [ADDM] [Year 2014] >
p <- ggplot(ASD_State_Subset, aes(x = reorder(State_Full1, Denominator, FUN =
y = Denominator)) +
  geom_bar(stat="identity", aes(fill = reorder(State_Full1, Denominator, FUN =
scale_fill_discrete(guide = guide_legend(title = "US. States")) + # Legend N
scale_x_discrete(name = "US. States") +
scale_y_continuous(name = "No. Children (Denominator)") +
ggtitle("No. Children Surveyed by State [ Source: ADDM ] [Year 2014]") +
# geom_text(aes(label=Denominator), vjust=1.6, color="darkslategrey", size=
theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),
axis.title = element_text(face = 'plain', color = "darkslategrey"),
legend.position="none")
```

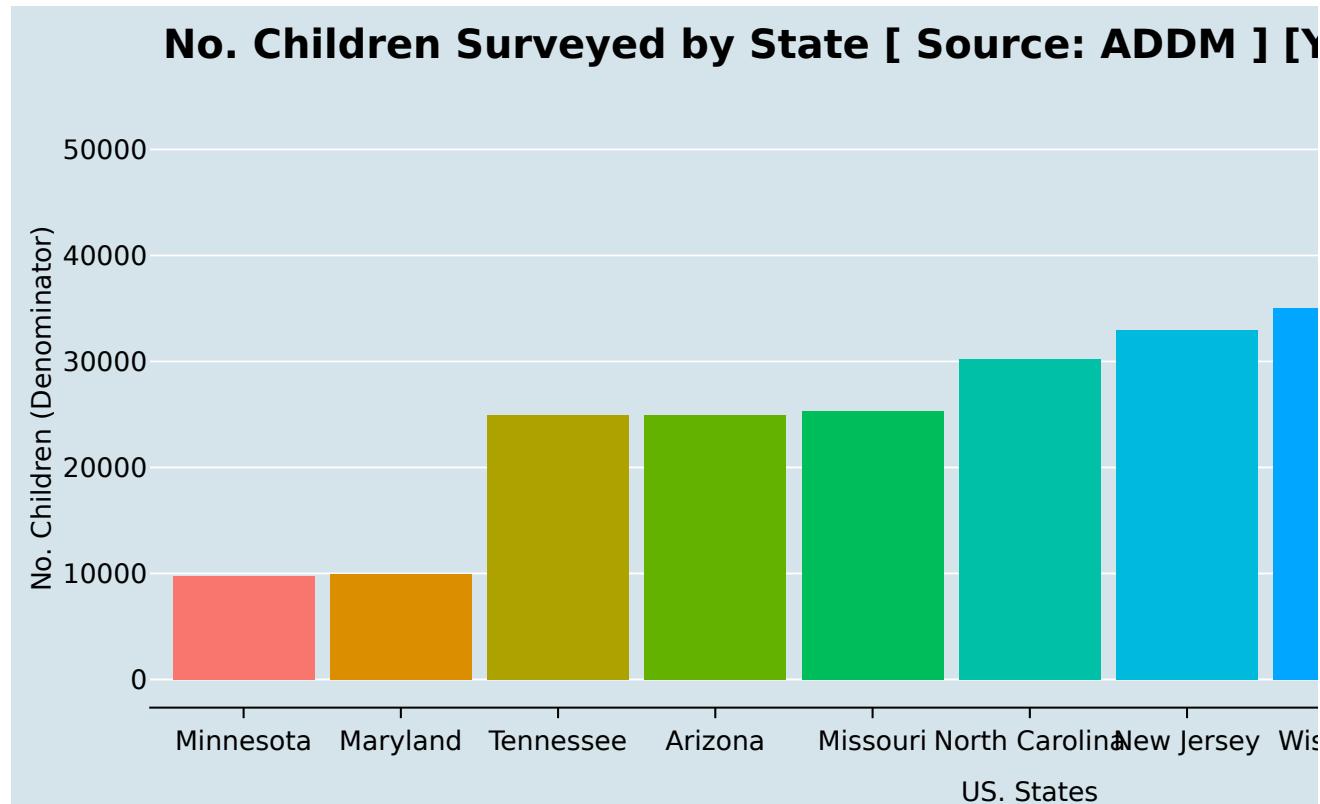
```
In [245]: # Show plot
p
```



```
In [246]: # Theme of the economist magazine:
# p + theme_economist() + scale_colour_economist() + theme(legend.position = '
```

In [247]: # Dynamic chart

```
p_dynamic <- p + theme_economist() + scale_colour_economist() + theme(legend.position = "none")
p_dynamic <- ggplotly(p_dynamic)
p_dynamic
```



Quiz:

Create No. Children Surveyed by State [Source: XXXX] [Year CCYY] for other data sources & years:

In [248]: # Write your code below and press Shift+Enter to execute

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

Quiz:

Create No. ASD Children by State [Source: XXXX] [Year CCYY] for other data sources & years:

Hint: Use variable: ASD_State_ADDM\$Numerator_ASD

In [249]: # Write your code below and press Shift+Enter to execute

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

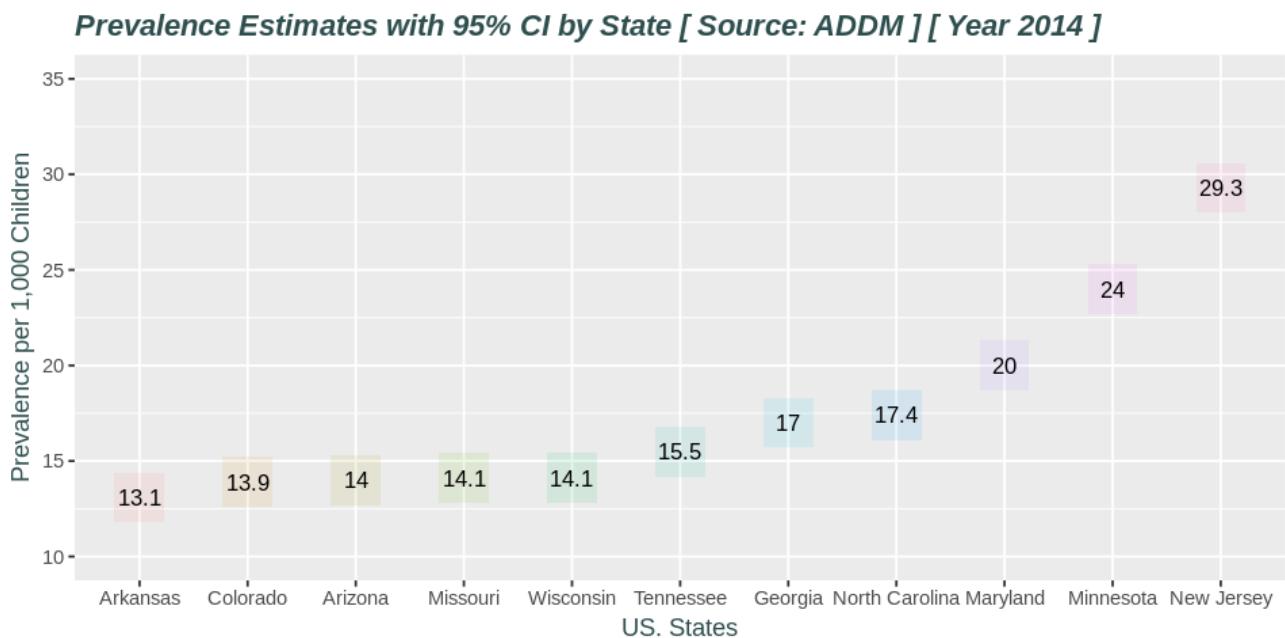
Data Visualisation (Enhanced) - [R] US. State Level Prevalence Estimates with 95% CI by State [Source: ADDM] [Year 2014]

```
In [250]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

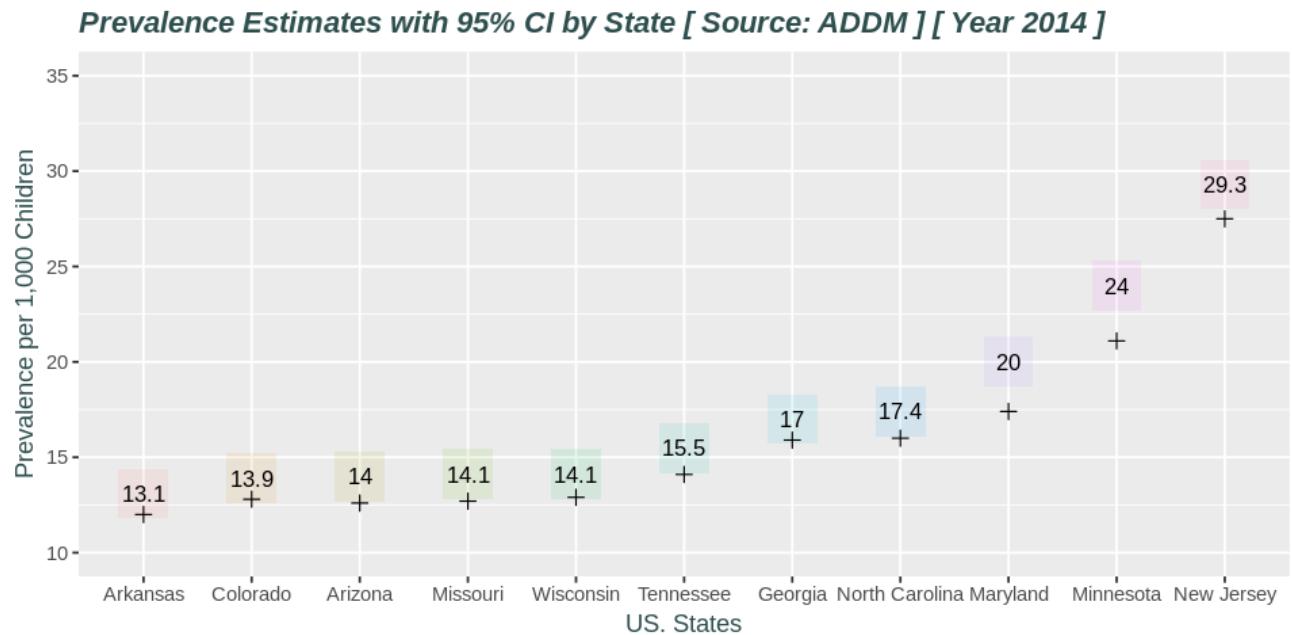
Visualise: Prevalence Estimates with 95% CI by State [Source: ADDM] [Year 2014]

```
In [251]: # ASD_State_Subset <- subset(ASD_State_ADDM, Year == 2014)  
# or  
# ASD_State_Subset <- subset(ASD_State, Source == 'addm' & Year == 2014)  
  
# Point plot/chart  
p = ggplot(ASD_State_Subset, aes(x = reorder(State_Full1, Prevalence, median),  
                                    y = Prevalence)) +  
    geom_point(stat="identity", aes(colour = reorder(State_Full1, Prevalence, me-  
                                         scale_colour_discrete(guide = guide_legend(title = "US. States")) + # Legend  
                                         scale_y_continuous(name = "Prevalence per 1,000 Children",  
                                                             breaks = seq(10, 35, 5),  
                                                             limits=c(10, 35)) +  
                                         scale_x_discrete(name = "US. States") +  
                                         gtitle("Prevalence Estimates with 95% CI by State [ Source: ADDM ] [ Year 2014 ]") +  
                                         theme(title = element_text(face = 'bold.italic', color = "darkslategrey"),  
                                               axis.title = element_text(face = 'plain', color = "darkslategrey"),  
                                               legend.position = 'none') +  
                                         geom_text(aes(label=Prevalence), hjust=0.5, color="black", size=3.5) # Show  
                                         # Add 95% CI bars  
                                         # Add state names below x-axis  
                                         # Add grid lines
```

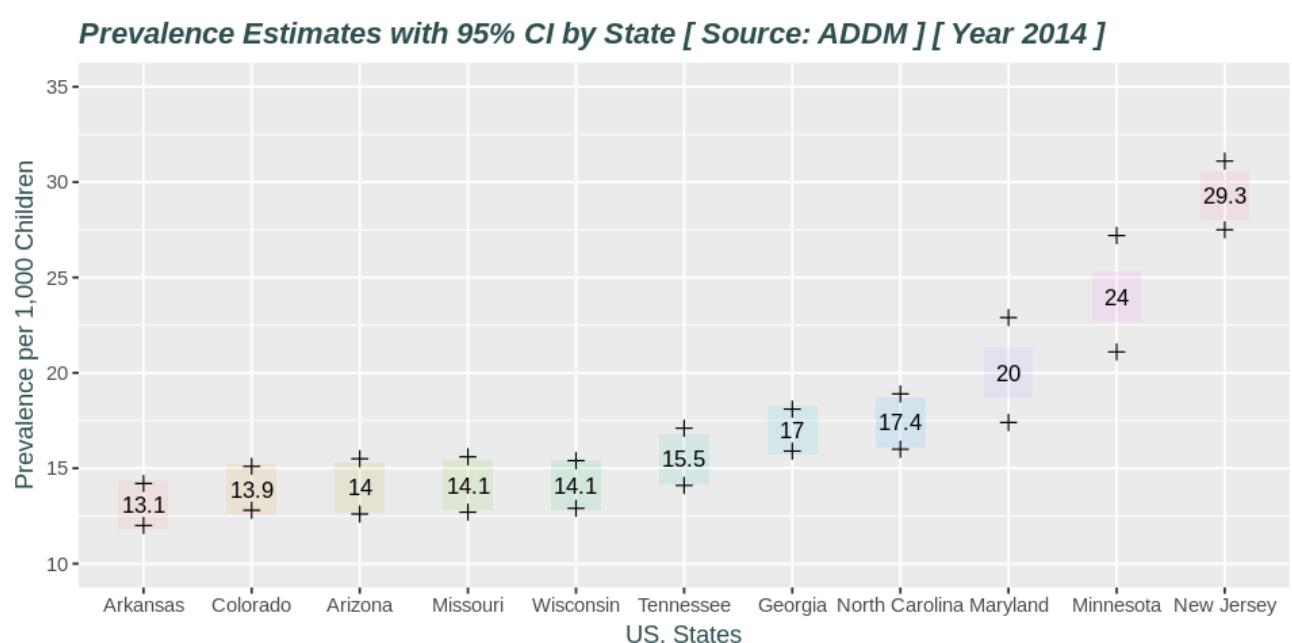
```
In [252]: # Show plot  
p
```



```
In [253]: # Add Lower.CI
p = p + geom_point(data = ASD_State_Subset, aes(x = reorder(State_Full1, Preva
                                         shape=Source # point shape
),
size = 2 # point size
) +
# geom_text(aes(label=Lower.CI), hjust=-0.1, vjust=3, color="darkslategrey",
scale_shape_manual(values=3) # manual define point shape
# Show plot
p
```



```
In [254]: # Add Upper.CI
p = p + geom_point(data = ASD_State_Subset, aes(x = reorder(State_Full1, Preva
                                         shape=Source # point shape
),
size = 2 # point size
)
# geom_text(aes(label=Upper.CI), hjust=-0.1, vjust=-3, color="darkslategrey",
# Show plot
p
```

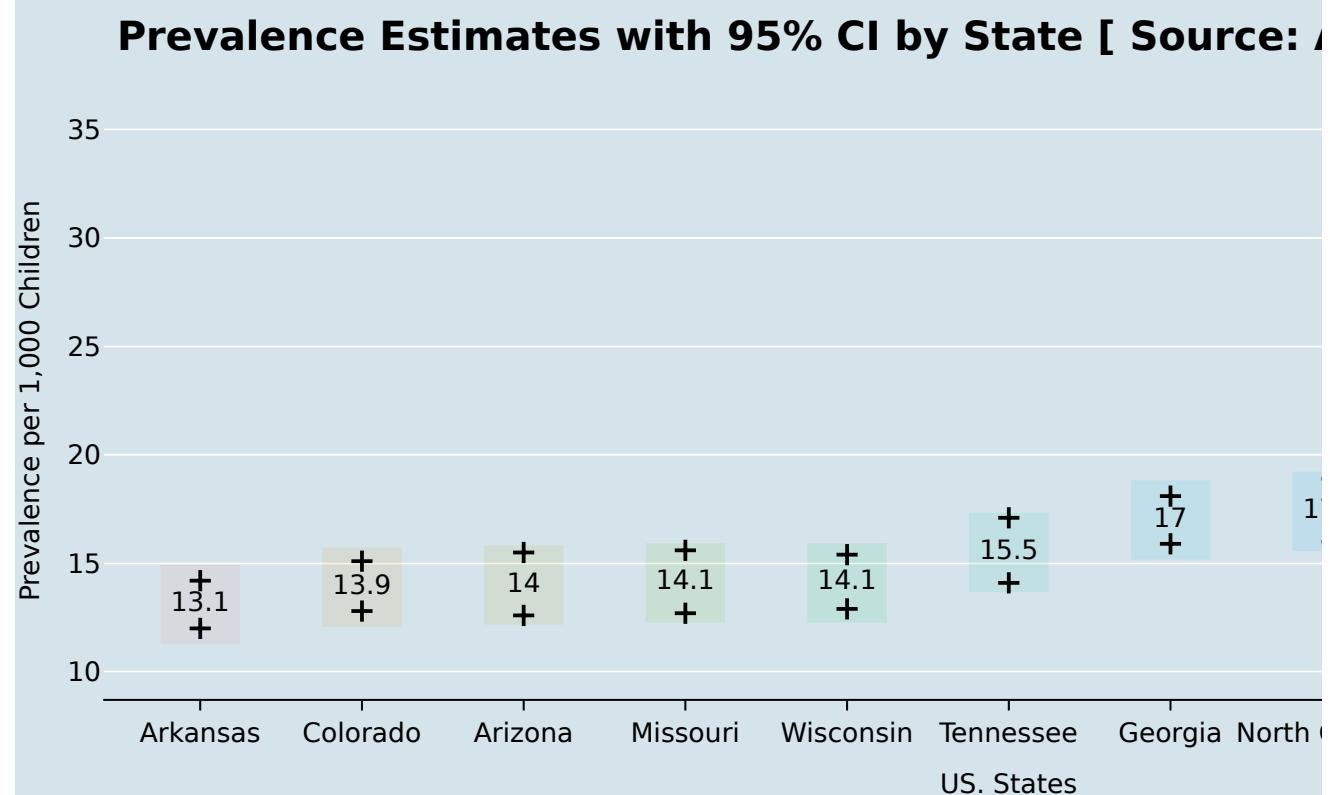


```
In [255]: # theme of the economist magazine:  
# p + theme_economist() + scale_colour_economist() + scale_colour_discrete(gui
```

```
In [256]: # Dynamic chart  
p_dynamic <- p + theme_economist() + scale_colour_economist() + scale_colour_d  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



Quiz:

Create Prevalence Estimates with 95% CI by State [Source: ADDM] [Year CCYY] for other data sources & years:

```
In [257]: # Write your code below and press Shift+Enter to execute
```

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

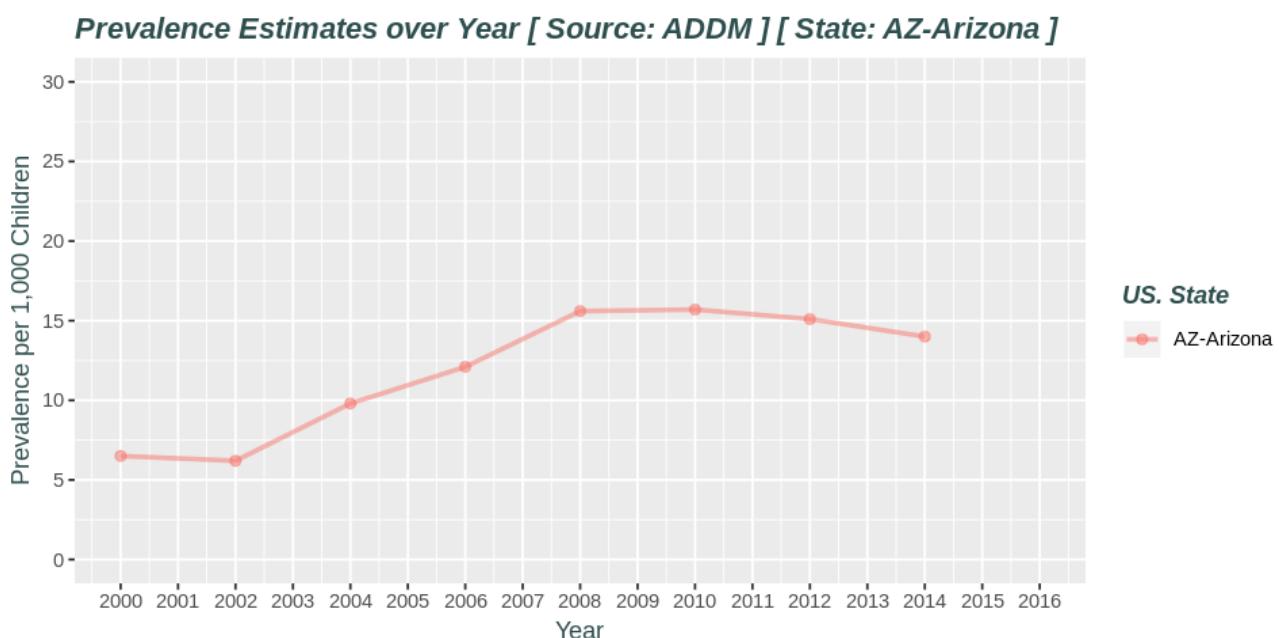
Data Visualisation (Enhanced) - [R] US. State Level Prevalence Estimates over Year [Source: ADDM] [State: AZ-Arizona]

```
In [258]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

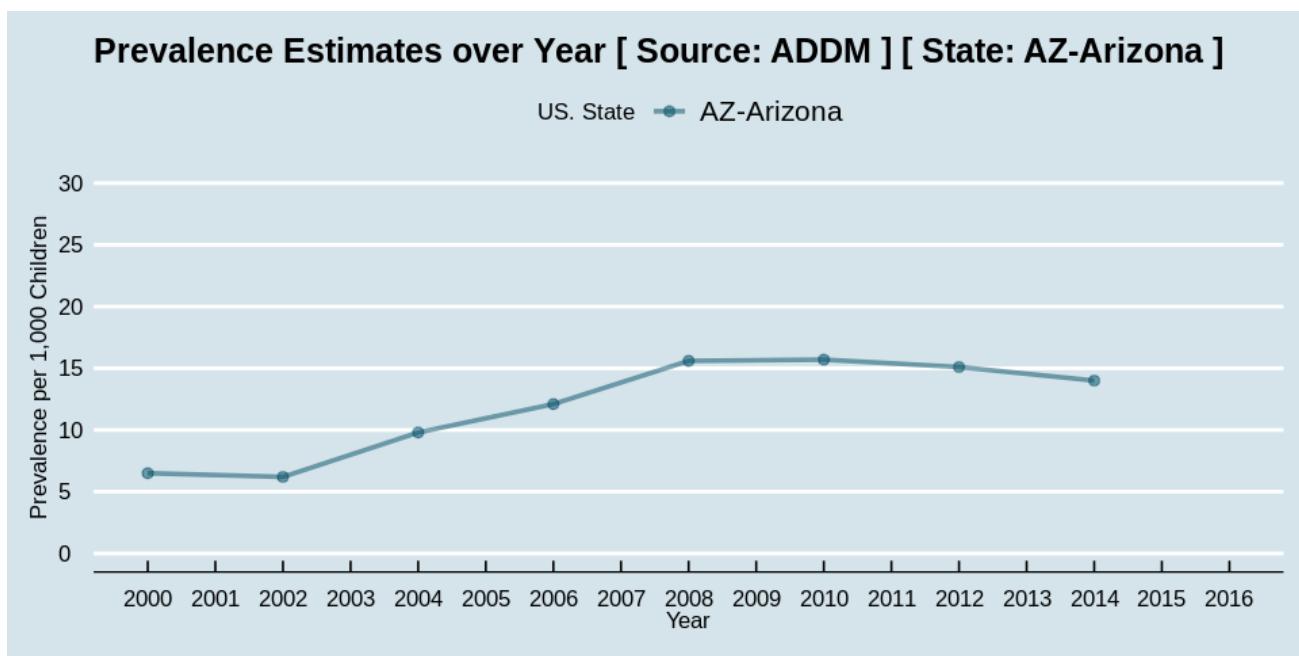
Visualise: Prevalence Estimates over Year [Source: ADDM] [State: AZ-Arizona]

```
In [259]: # All year/time Prevalence data with: Source_UC == 'ADDM' & State_Full2 == 'AZ'  
ASD_State_Subset <- subset(ASD_State, Source_UC == 'ADDM' & State_Full2 == 'AZ')  
  
# Line plot/chart for < State ASD Prevalence [ADDM] [AZ-Arizona] >  
p <- ggplot(ASD_State_Subset, aes(x = Year, y = Prevalence))  
# Select (add) line chart type:  
p <- p + geom_line(aes(color = State_Full2),  
                    linetype = "solid", # http://sape.inf.usi.ch/quick-referen  
                    size=1,  
                    alpha=0.5)  
# Select (add) points to chart:  
p <- p + geom_point(aes(color = State_Full2),  
                     size=3,  
                     shape=20,  
                     alpha=0.5)  
# Customize legend name:  
p <- p + labs(color = "US. State")  
# Adjust x and y axis, scale, limit and labels:  
p <- p + scale_y_continuous(name = "Prevalence per 1,000 Children",  
                            breaks = seq(0, 30, 5),  
                            limits=c(0, 30)) +  
    scale_x_continuous(name = "Year",  
                      breaks = seq(2000, 2016, 1),  
                      limits = c(2000, 2016))  
# Customize chart title:  
p <- p + ggtitle("Prevalence Estimates over Year [ Source: ADDM ] [ State: AZ-  
# Customize chart title and axis labels:  
p <- p + theme(title = element_text(face = 'bold.italic', color = "darkslategr  
axis.title = element_text(face = 'plain', color = "darkslategre
```

```
In [260]: # Show plot  
p
```



```
In [261]: # Theme of the economist magazine:  
p + theme_economist() + scale_colour_economist()
```



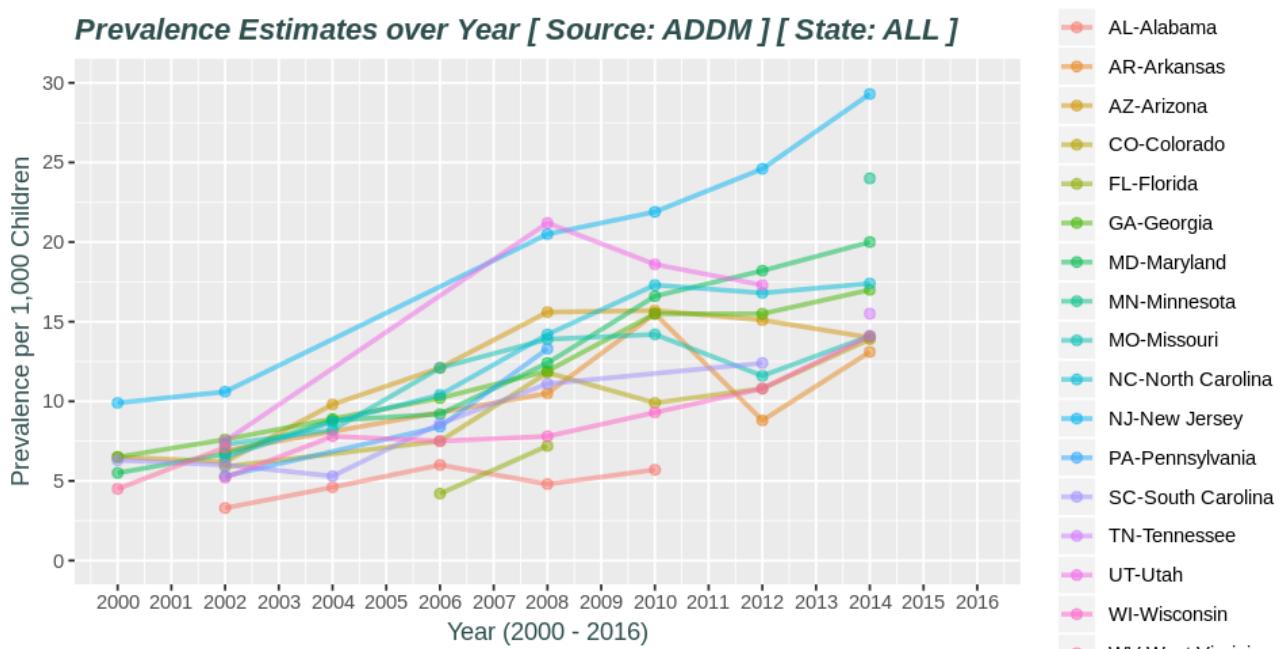
Data Visualisation (Enhanced) - [R] US. State Level Prevalence Estimates over Year [Source: ADDM] [State: ALL]

```
In [262]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

Visualise: Prevalence Estimates over Year [Source: ADDM] [State: ALL]

```
In [263]: p <- ggplot(ASD_State_ADDM, aes(x = Year, y = Prevalence))
# Select (add) line chart type:
p <- p + geom_line(aes(color = State_Full2),
                     linetype = "solid", # http://sape.inf.usi.ch/quick-referen
                     size=1,
                     alpha=0.5)
# Select (add) points to chart:
p <- p + geom_point(aes(color = State_Full2),
                     size=3,
                     shape=20,
                     alpha=0.5)
# Show plot
# p
# Customize line color and legend name:
p <- p + labs(color = "US. State")
# Adjust x and y axis, scale, limit and labels:
p <- p + scale_y_continuous(name = "Prevalence per 1,000 Children",
                             breaks = seq(0, 30, 5),
                             limits=c(0, 30)) +
  scale_x_continuous(name = "Year (2000 - 2016)",
                     breaks = seq(2000, 2016, 1),
                     limits = c(2000, 2016))
# Customize chart title:
p <- p + ggtitle("Prevalence Estimates over Year [ Source: ADDM ] [ State: ALL ]")
# Customize chart title and axis labels:
p <- p + theme(title = element_text(face = 'bold.italic', color = "darkslategray"),
                axis.title = element_text(face = 'plain', color = "darkslategray"),
                legend.position="right")
```

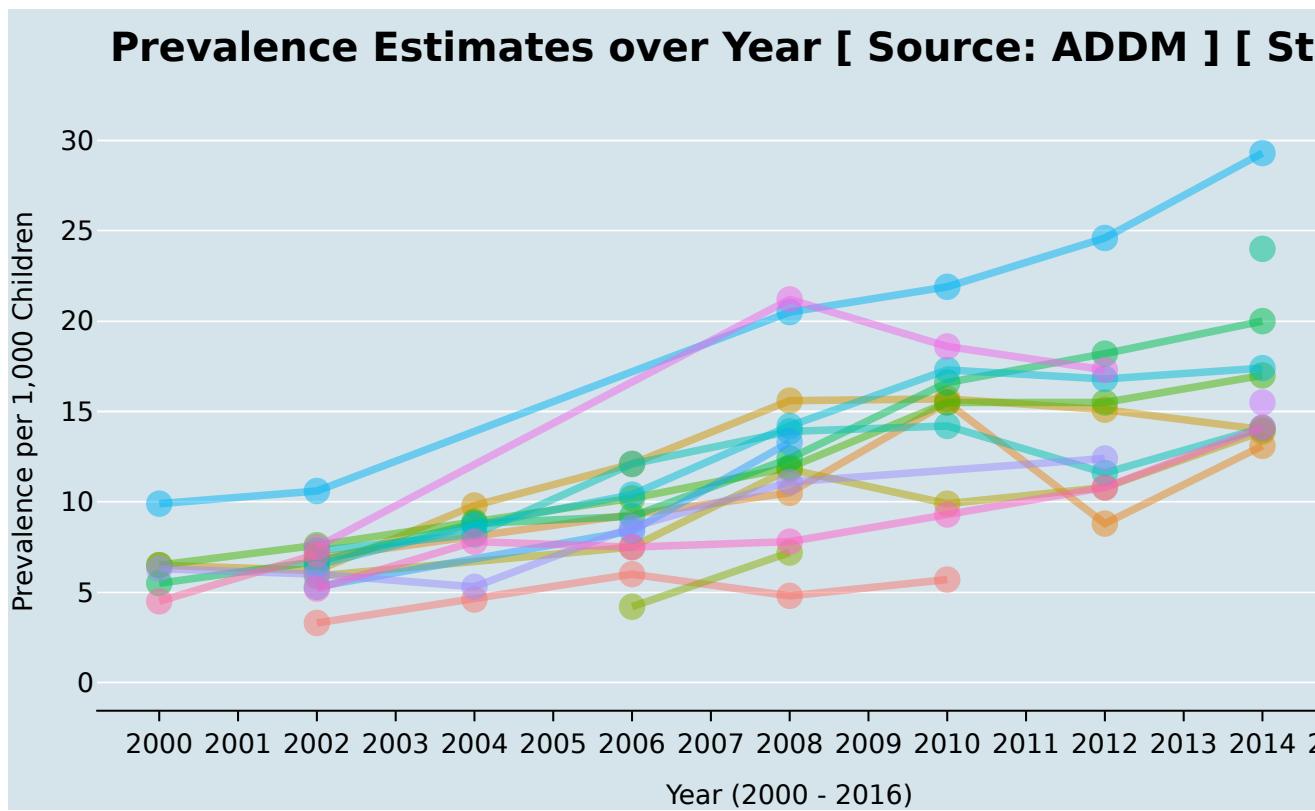
```
In [264]: # Show plot
p
```



In [265]: # Dynamic chart

```
p_dynamic <- p + theme_economist() + scale_colour_economist() + scale_colour_d  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```

Scale for 'colour' is already present. Adding another scale for 'colour', which will replace the existing scale.



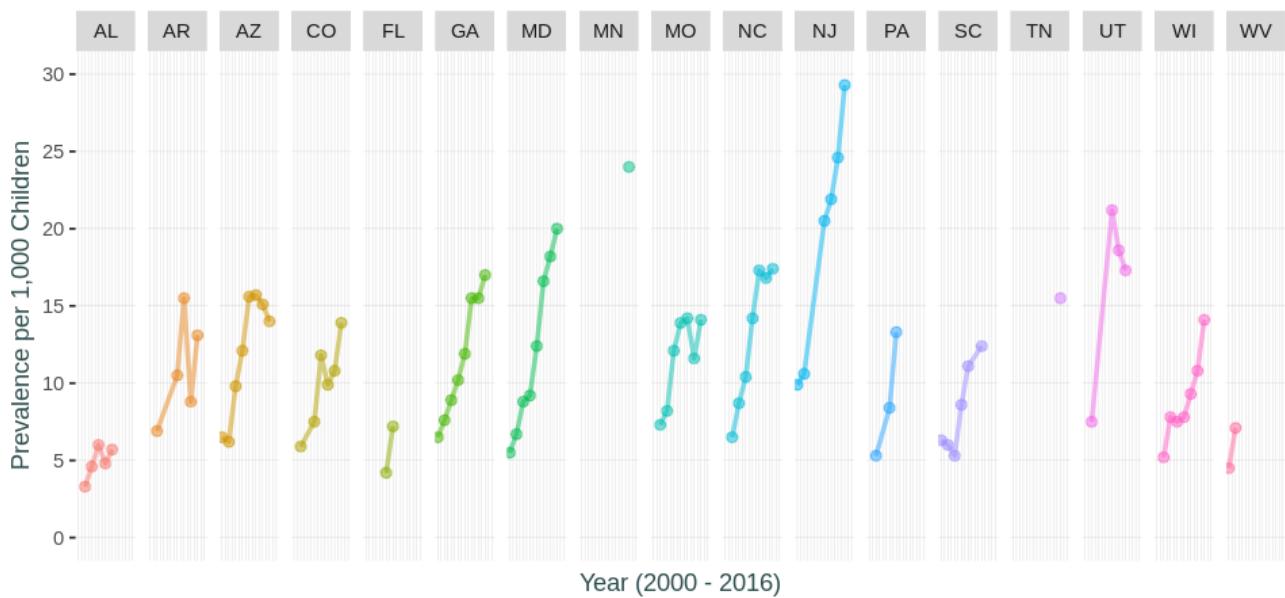
Split chart by state

```
In [266]: # Show plot in facet_grid
p + facet_grid(facets = . ~ State) +
  theme(legend.position = "none", # Hide legend
        axis.text.x=element_blank(), # Hide axis
        axis.ticks.x=element_blank(), # Hide axis
        panel.background = element_blank(), # Remove panel background
        panel.grid.major = element_line(size = 0.1, linetype = 1, colour = "lightblue"))
)
```

geom_path: Each group consists of only one observation. Do you need to adjust the group aesthetic?

geom_path: Each group consists of only one observation. Do you need to adjust the group aesthetic?

Prevalence Estimates over Year [Source: ADDM] [State: ALL]

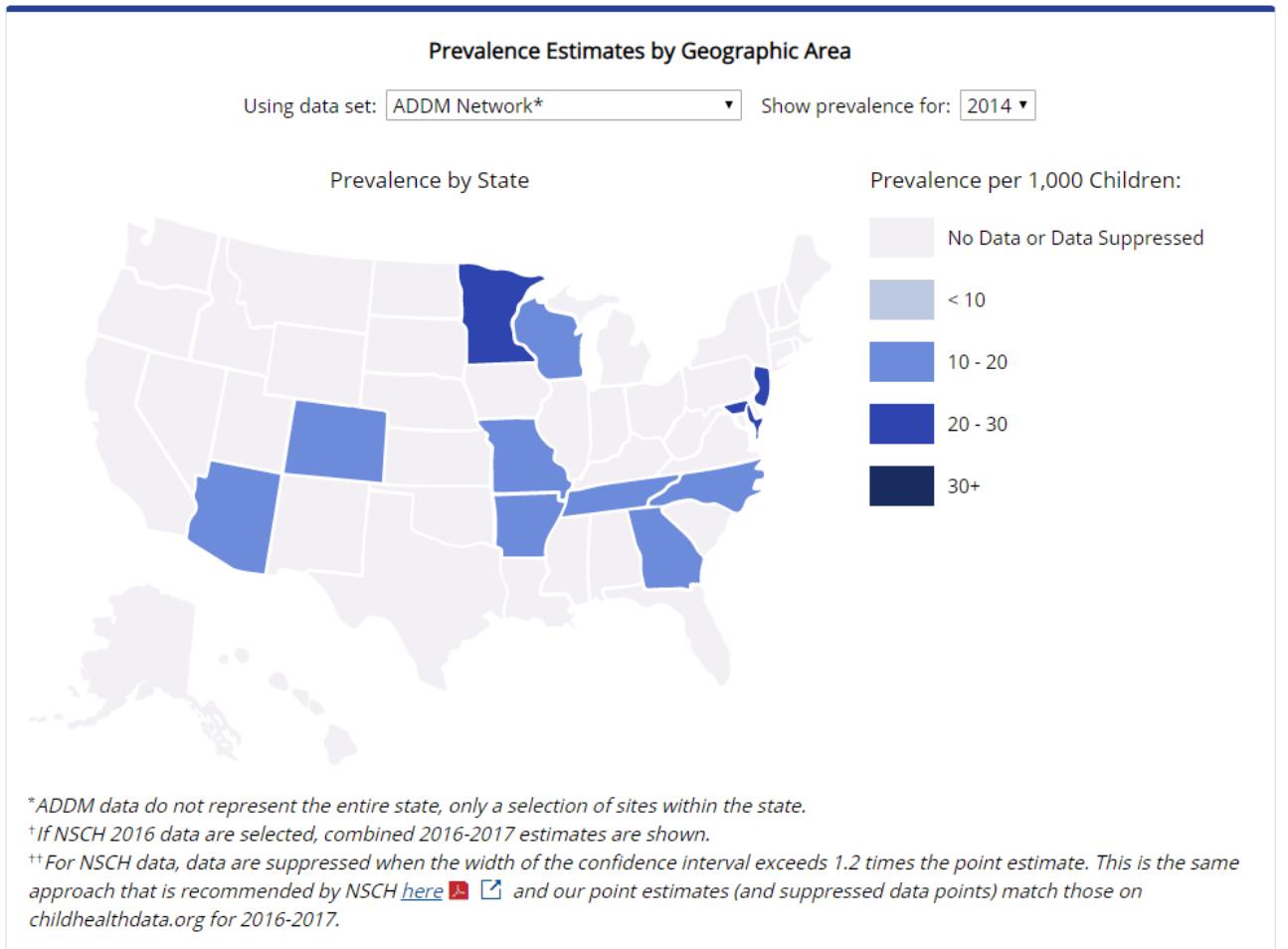


Data Visualisation (Enhanced) - Plotting on Map

```
In [267]: # -----
# EDA - Visualisation on map
# -----
if(!require(usmap)){install.packages("usmap")}
library(usmap) # usmap: Mapping the US
```

Loading required package: usmap

Data Visualisation (Enhanced) - Plotting on Map [CDC] REPORTED PREVALENCE VARIES BY GEOGRAPHIC LOCATION



Data Visualisation (Enhanced) - Plotting on Map [R] REPORTED PREVALENCE VARIES BY GEOGRAPHIC LOCATION

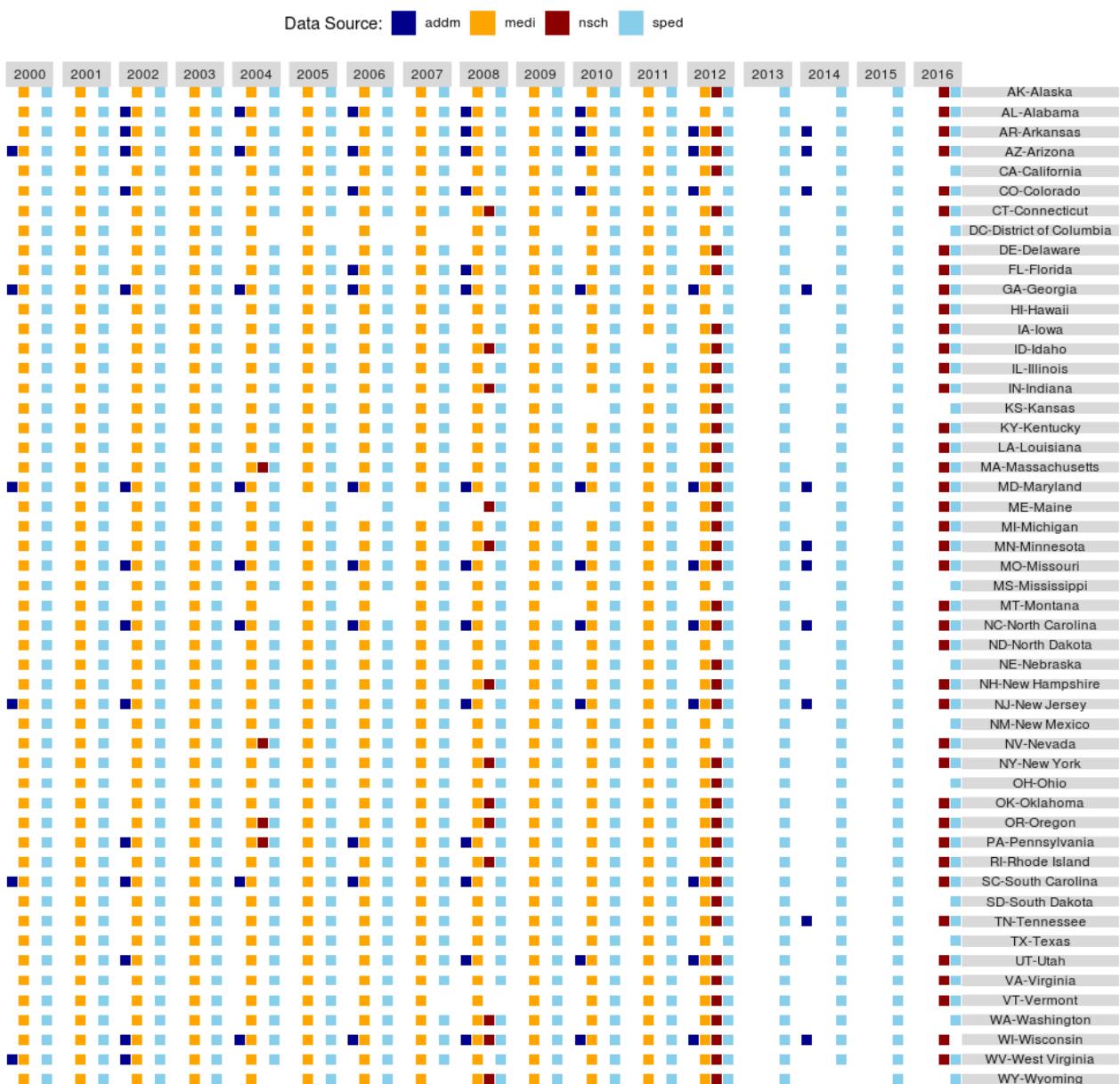
Let's review data availability by data Sources & Years:

- ASD_State_ADDM in Years: 2000, 2002, 2004, 2006, 2008, 2010, 2012, 2014
- ASD_State_MEDI in Years: 2000 ~ 2012
- ASD_State_NSCH in Years: 2004, 2008, 2012, 2016
- ASD_State_SPED in Years: 2000 ~ 2016

Years Data Available



Years Data Available by State



Data Visualisation (Enhanced) - Plotting on Map [R] REPORTED PREVALENCE VARIES BY GEOGRAPHIC LOCATION [Source: ADDM] [Year: 2014]

```
In [268]: # Adjust in-line plot size to M x N
# options(repr.plot.width=8, repr.plot.height=4)
```

Prepare US State level data: [Source: ADDM] [Year: 2014]

```
In [269]: # Prepare data - addm 2014
Map_Data_Source = 'addm' # Available values lowercase: 'addm', 'medi', 'nsch',
Map_Data_Value = 'Prevalence' # variable must be numeric, variable name in 'qu

# Uncomment below to use Prevalence of different groups:
# Map_Data_Value = 'Male.Prevalence' # variable must be numeric, variable name
# Map_Data_Value = 'Female.Prevalence' # variable must be numeric, variable na
# Map_Data_Value = 'Asian.or.Pacific.Islander.Prevalence' # variable must be n

Map_Data_Year = 2014 # must be integer
ASD_State_Subset = subset(ASD_State, Source == Map_Data_Source & Year == Map_D
```

The usmap package/function requires input data to have a column of **state**, or **fips**. (case sensitive)

- state: Name of US state
- fips: FIPS code for either a US state

<https://cran.r-project.org/web/packages/usmap/vignettes/mapping.html> (<https://cran.r-project.org/web/packages/usmap/vignettes/mapping.html>)

<https://cran.r-project.org/web/packages/usmap/usmap.pdf> (<https://cran.r-project.org/web/packages/usmap/usmap.pdf>)

```
In [270]: # The usmap package/function requires input data to have a column of 'state',
ASD_State_Subset$state = ASD_State_Subset$State
# Glance
head(ASD_State_Subset)
```

State	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Source	Source_Full1	State_Full1	State_F
AZ	24952	14.0	12.6	15.5	2014	addm	Autism & Developmental Disabilities Monitoring Network	Arizona	AZ-Arizo
AR	39992	13.1	12.0	14.2	2014	addm	Autism & Developmental Disabilities Monitoring Network	Arkansas	Arkans
CO	41128	13.9	12.8	15.1	2014	addm	Autism & Developmental Disabilities Monitoring Network	Colorado	Colo

Visualise: **Prevalence Estimates by Geographic Area** [Source: ADDM] [Year: 2014]

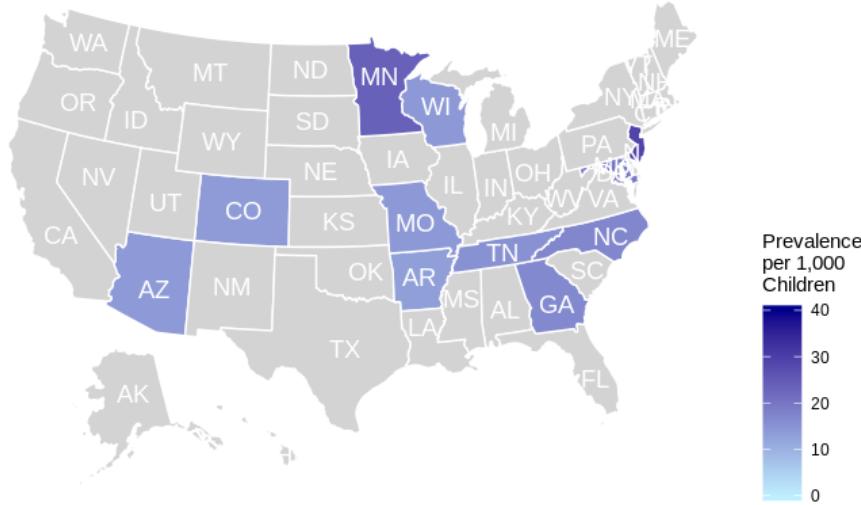
In [271]: # Show data on map

```
p_map_addm_2014 <- plot_usmap(data = ASD_State_Subset, values = Map_Data_Value
                                color = "white", # map line colour
                                labels = TRUE, # State name shown
                                label_color = 'white' # State name colour
) +
  scale_fill_continuous(
    na.value = "lightgrey", # Set colour with no State data
    low="lightblue", high = "darkblue",
    name = "Prevalence\\nper 1,000\\nChildren",
    limits=c(0, 40) #same colour levels/limits for plots
) +
  labs(title = paste("Prevalence Estimates by Geographic Area", '\n[ Measure :',
                     subtitle = 'https://www.cdc.gov/ncbdd/autism'
) +
  theme(panel.background = element_rect(color = "white", fill = "white"),
        legend.position = "right")
```

Show map

```
p_map_addm_2014
```

Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : addm] [Year : 2014]
<https://www.cdc.gov/ncbdd/autism>

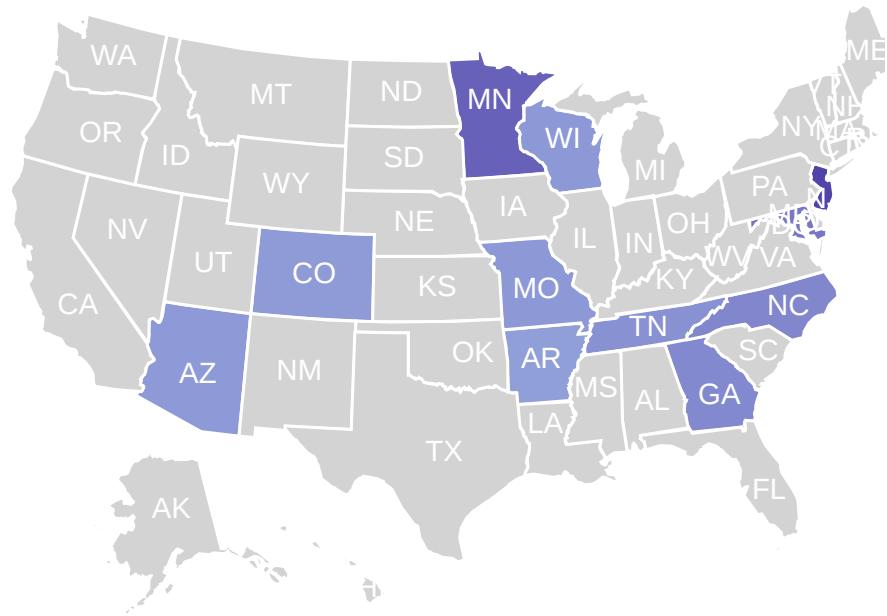


In [272]: # Dynamic map

```
p_dynamic <- p_map_addm_2014  
p_dynamic <- ggplotly(p_dynamic)  
p_dynamic
```

Prevalence Estimates by Geographic Area

[Measure : Prevalence] [Source : addm] [Year : 2014]



Data Visualisation (Enhanced) - Plotting on Map [R] REPORTED PREVALENCE VARIES BY GEOGRAPHIC LOCATION [Source: NSCH] [Year: 2004, 2008, 2012, 2016]

Prepare US State level data: [Source: NSCH] [Year: ALL]

In [273]:

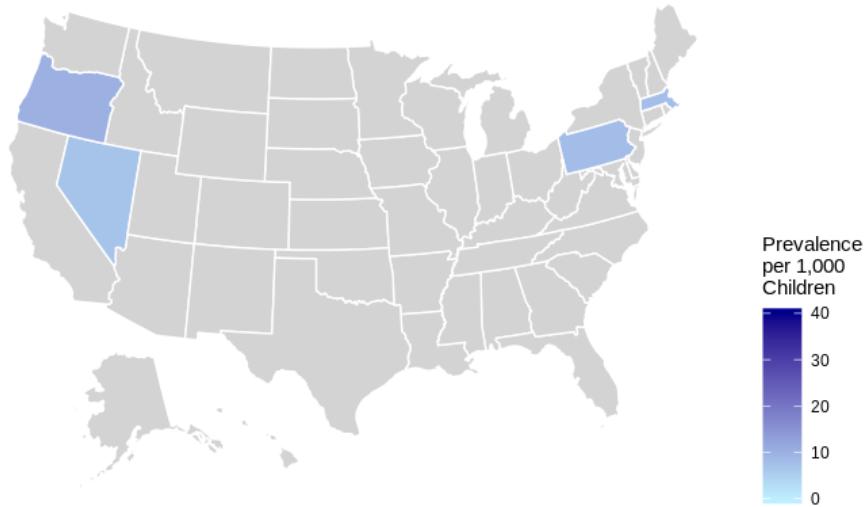
```
Map_Data_Source = 'nsch' # Available values lowercase: 'addm', 'medi', 'nsch'  
Map_Data_Value = 'Prevalence' # variable must be numeric, variable name in 'qu
```

Visualise: Prevalence Estimates by Geographic Area [Source: NSCH] [Year: 2004]

In [274]: # Prepare data - nsch 2004

```
Map_Data_Year = 2004 # must be integer
ASD_State_Subset = subset(ASD_State, Source == Map_Data_Source & Year == Map_D
ASD_State_Subset$state = ASD_State_Subset$State
# Plot on map
p_map_nsch_2004 <- plot_usmap(data = ASD_State_Subset, values = Map_Data_Value
p_map_nsch_2004
```

Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2004]
<https://www.cdc.gov/ncbddd/autism>

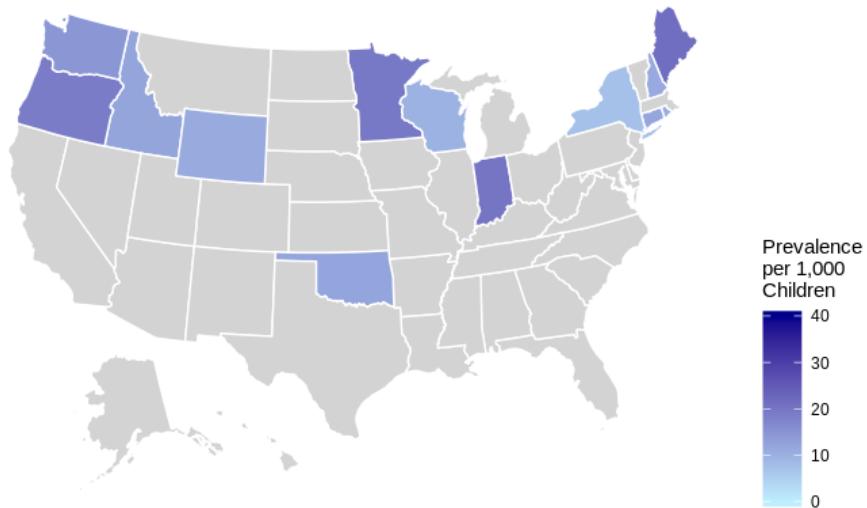


Visualise: Prevalence Estimates by Geographic Area [Source: NSCH] [Year: 2008]

In [275]: # Prepare data - nsch 2008

```
Map_Data_Year = 2008 # must be integer
ASD_State_Subset = subset(ASD_State, Source == Map_Data_Source & Year == Map_D
ASD_State_Subset$state = ASD_State_Subset$State
p_map_nsch_2008 <- plot_usmap(data = ASD_State_Subset, values = Map_Data_Value
p_map_nsch_2008
```

Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2008]
<https://www.cdc.gov/ncbddd/autism>

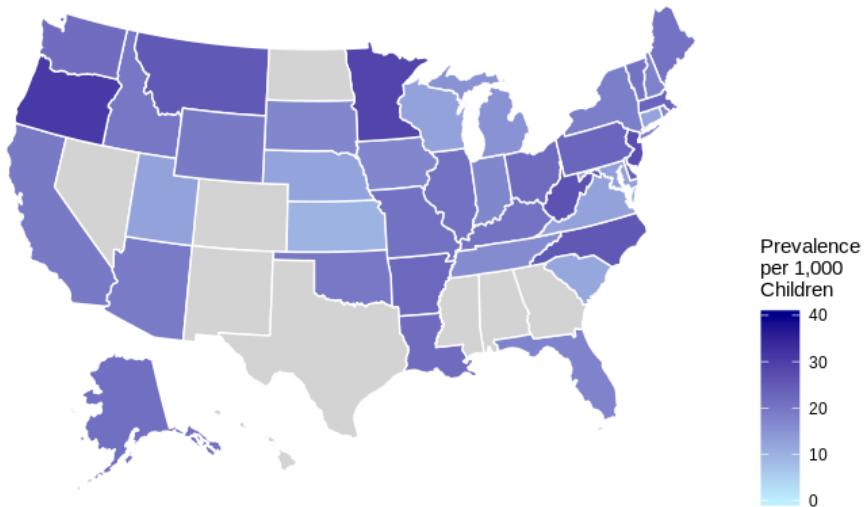


Visualise: Prevalence Estimates by Geographic Area [Source: NSCH] [Year: 2012]

In [276]: # Prepare data - nsch 2012

```
Map_Data_Year = 2012 # must be integer
ASD_State_Subset = subset(ASD_State, Source == Map_Data_Source & Year == Map_D
ASD_State_Subset$state = ASD_State_Subset$State
p_map_nsch_2012 <- plot_usmap(data = ASD_State_Subset, values = Map_Data_Value
p_map_nsch_2012
```

Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2012]
<https://www.cdc.gov/ncbddd/autism>

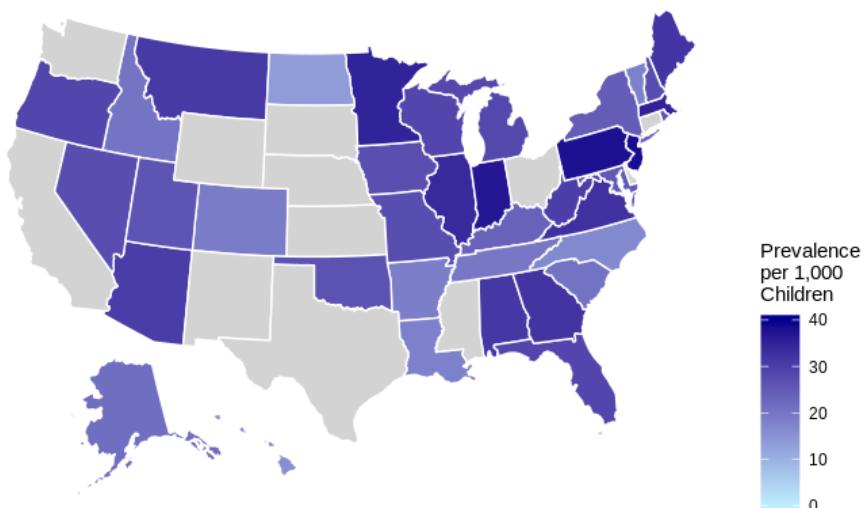


Visualise: Prevalence Estimates by Geographic Area [Source: NSCH] [Year: 2016]

In [277]: # Prepare data - nsch 2016

```
Map_Data_Year = 2016 # must be integer
ASD_State_Subset = subset(ASD_State, Source == Map_Data_Source & Year == Map_D
ASD_State_Subset$state = ASD_State_Subset$State
p_map_nsch_2016 <- plot_usmap(data = ASD_State_Subset, values = Map_Data_Value
p_map_nsch_2016
```

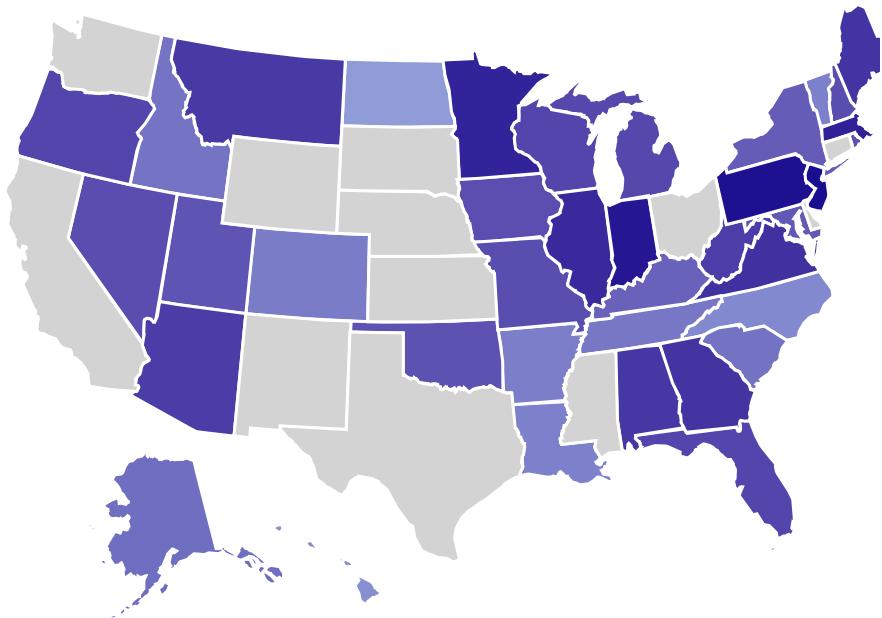
Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2016]
<https://www.cdc.gov/ncbddd/autism>



In [278]: # Dynamic map

```
p_dynamic <- p_map_nsch_2016 # [ Source: NSCH ] [ Year: 2016 ]
p_dynamic <- ggplotly(p_dynamic)
p_dynamic
```

Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2016]



Combine multiple plots to show in one page/screen:

In [279]: # Adjust in-line plot size to M x N

```
options(repr.plot.width=8, repr.plot.height=6)
```

In [280]:

```
# -----  
# Combine multiple plots  
# -----  
if(!require(cowplot)){install.packages("cowplot")}  
library('cowplot')  
cowplot:::plot_grid(  
  p_map_nsch_2004,  
  p_map_nsch_2008,  
  p_map_nsch_2012,  
  p_map_nsch_2016,  
  nrow = 2)
```

Loading required package: cowplot

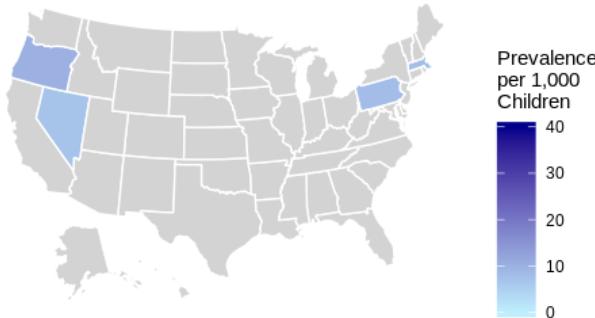
```
*****  
Note: As of version 1.0.0, cowplot does not change the  
default ggplot2 theme anymore. To recover the previous  
behavior, execute:  
theme_set(theme_cowplot())  
*****
```

Attaching package: 'cowplot'

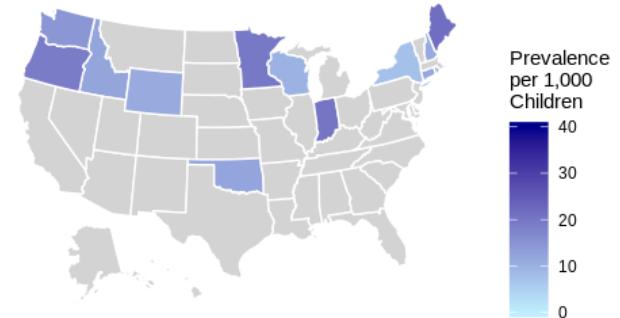
The following object is masked from 'package:ggthemes':

theme_map

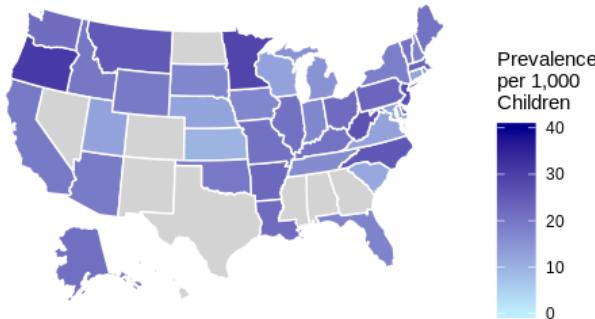
Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2004]
<https://www.cdc.gov/ncbddd/autism>



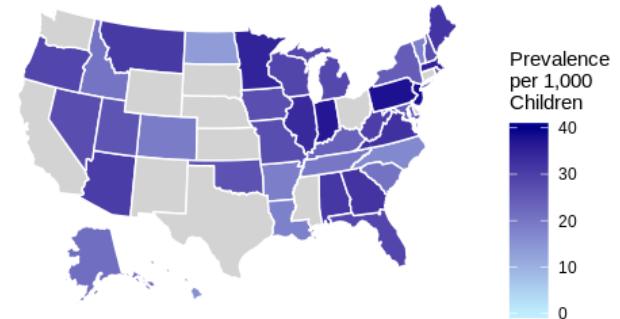
Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2008]
<https://www.cdc.gov/ncbddd/autism>



Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2012]
<https://www.cdc.gov/ncbddd/autism>



Prevalence Estimates by Geographic Area
[Measure : Prevalence] [Source : nsch] [Year : 2016]
<https://www.cdc.gov/ncbddd/autism>



Export current plot as image file:

```
In [281]: # -----  
# Export current plot as image file  
# -----  
ggsave("plot Map Prevalence Estimates by Geographic Area [NSCH] [2004-2016].png",  
       width = 60, height = 30, units = 'cm')
```

.0

Sampling & Normality

Sampling & Normality - Population & Sample

```
In [282]: # Adjust in-line plot size to M x N  
# options(repr.plot.width=8, repr.plot.height=4)
```

Create a **Population** of US. State level ASD Prevalence from Source SPED in Year 2016

```
In [283]: # -----  
# Create a *Population* of US. State level ASD Prevalence from Source SPED in  
# -----  
ASD_State_SPED_2016 <- subset(ASD_State, Source == 'sped' & Year == 2016, select = c(State, Prevalence))  
head(ASD_State_SPED_2016)
```

State	Prevalence
AL	9.1
AK	10.1
AZ	10.4
AR	9.5
CA	13.9
CO	7.3

```
In [284]: dim(ASD_State_SPED_2016)  
# *Population* mean Prevalence  
mean(ASD_State_SPED_2016$Prevalence)
```

50 2

11.182

Define a function to calculate population std-dev (Omega):

```
In [285]: # Use sd() to calculate *sample* std-dev (S)
# Use sd.p() to calculate *population* std-dev (Omega)

# Define a function sd.p() to calculate *population* std-dev (Omega)
# https://www.dummies.com/education/math/statistics/standard-deviation-r/

sd.p = function(x) {sd(x) * sqrt((length(x)-1)/length(x))}

# Treat as sample:
cat('sd() of ASD_State_SPED_2016$Prevalence : ', sd(ASD_State_SPED_2016$Prev

# Treat as population:
cat('\nsd.p() of ASD_State_SPED_2016$Prevalence : ', sd.p(ASD_State_SPED_2016$

sd() of ASD_State_SPED_2016$Prevalence : 3.233226
sd.p() of ASD_State_SPED_2016$Prevalence : 3.200731
```

Sampling & Normality - Central Limit Theorem (CLT)

Create a **Sample** of US. State level ASD Prevalence from Source SPED in Year 2016

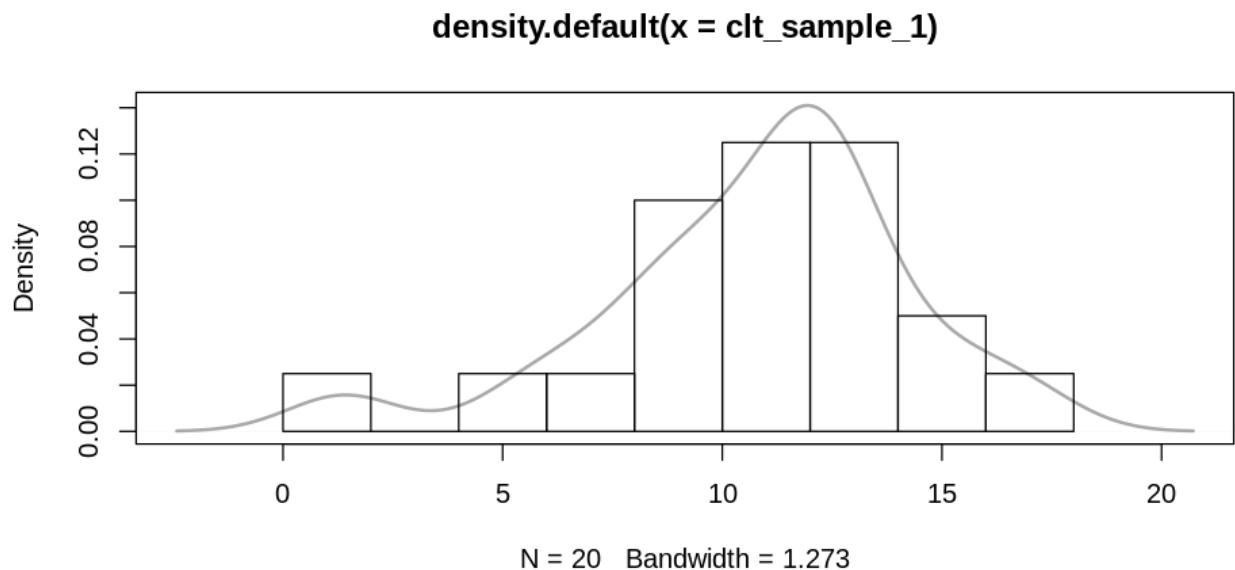
```
In [286]: # Create a *Sample* from ASD_State_SPED_2016$Prevalence,
# with sample size n =
clt_n = 20
# clt_n = 40

set.seed(88)
clt_sample_1 = sample(x = ASD_State_SPED_2016$Prevalence, size = clt_n, replace = TRUE)
clt_sample_1
```

11.2 9.5 16.9 6.9 11.2 8.2 12.7 9.5 1.4 12.7 10.1 14.2 11.9 11.9 8.5 12.1 12.1
15.4 13 5.5

```
In [287]: # Adjust in-line plot size to M x N
options(repr.plot.width=8, repr.plot.height=4)
```

```
In [288]: plot(density(clt_sample_1), col="darkgrey", lwd=2)
hist(clt_sample_1, probability = T, add = T)
```



Draw a large k number of samples, with sample size = n:

```
In [289]: # Repeatedly sample for k times, create a matrix/array to store these samples
clt_k = 10000 # or called 'N', but this can be confusing due to N can also be
set.seed(88) # Repeatable sampling using pseudo random method
clt_sample_k <- (replicate(clt_k, sample(x = ASD_State_SPED_2016$Prevalence, s
# first few samples
clt_sample_k[, 1:6]
```

```
11.2   9.5  11.2   8.3   8.3   6.9
 9.5  11.9  10.4  11.9  11.9  15.2
16.9   6.9   9.3   1.4  15.4  19.4
 6.9  13.0  19.4  12.7   8.5  15.1
11.2   8.3  12.1   9.1  11.2  12.1
 8.2  12.1  10.1  14.2  10.2   1.4
12.7  10.8  12.1   8.6   9.5  15.1
 9.5  15.2  14.1  14.1  10.8   6.9
 1.4   8.7  12.5  16.7   9.8   9.5
12.7   9.0   5.5  13.9  19.4  14.2
10.1   9.3   9.5   9.5  14.2  19.4
14.2  12.1  16.7  12.5  14.1  16.9
11.9   1.4   9.6  10.8  11.9   9.6
```

```
In [290]: # last sample
clt_sample_k[, clt_k]
```

```
8.6 14.1 12.1 19.4 12.7 10.3 10.3 11.2 10.4 9 10.4 13.9 9.3 14.2 11 12.1 9.5
10.1 13 15.2
```

```
In [291]: # mean values of first few samples
```

```
mean(clt_sample_k[, 1])
mean(clt_sample_k[, 2])
mean(clt_sample_k[, 3])
mean(clt_sample_k[, 4])
mean(clt_sample_k[, 5])
mean(clt_sample_k[, 6])

# or use apply() function to loop
apply(clt_sample_k[, 1:6], 2, mean)
```

10.745

9.94

12.705

10.67

12.045

12.795

10.745 9.94 12.705 10.67 12.045 12.795

```
In [292]: # std-dev values of first few samples
```

```
sd(clt_sample_k[, 1])
sd(clt_sample_k[, 2])
sd(clt_sample_k[, 3])
sd(clt_sample_k[, 4])
sd(clt_sample_k[, 5])
sd(clt_sample_k[, 6])

# or use apply() function to loop
apply(clt_sample_k[, 1:6], 2, sd)
```

3.52158382430107

3.7529497170822

3.57115269611191

4.01026314926038

3.17895630009202

5.01140541270873

3.52158382430107 3.7529497170822 3.57115269611191 4.01026314926038

3.17895630009202 5.01140541270873

k sample's distributions (k many)

In [293]:

```
# -----  
# k sample's distributions (k many)  
# -----  
# Show the first few sample's histogram  
par(mfrow=c(2, 3))  
apply(clt_sample_k[, 1:6], 2, FUN=hist)  
# Reset  
par(mfrow=c(1, 1))  
  
[[1]]  
$breaks  
[1] 0 2 4 6 8 10 12 14 16 18  
  
$counts  
[1] 1 0 1 1 4 5 5 2 1  
  
$density  
[1] 0.025 0.000 0.025 0.025 0.100 0.125 0.125 0.050 0.025  
  
$mids  
[1] 1 3 5 7 9 11 13 15 17  
  
$xname  
[1] "newX[, i]"  
  
$equidist  
[1] TRUE
```

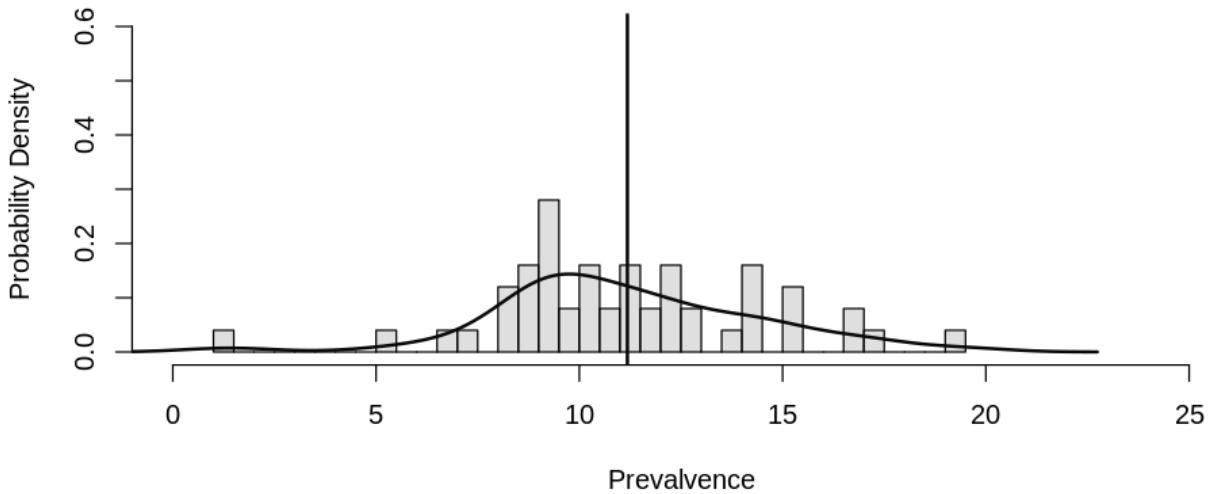
Show the first few sample's density, together with Population

```
In [294]: # Show the first few sample's density, together with Population
# Population (Prevalence) histogram in probability
hist(ASD_State_SPED_2016$Prevalence, probability = T,
      col=rgb(0.75,0.75,0.75,0.5), breaks = 50,
      xlab = 'Prevalence', xlim = (c(0, 25)),
      ylab = 'Probability Density',
      ylim = (c(0, 0.6)),
      main = 'Visualize Population & Samples')

# Overlay curve:
# Population (Prevalence) density
lines(density(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)

# Overlay line:
# mean = mean of Population (Prevalence)
abline(v=mean(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)
```

Visualize Population & Samples



In [295]:

```

# Show the first few sample's density, together with Population
# Population (Prevalence) histogram in probability
hist(ASD_State_SPED_2016$Prevalence, probability = T,
      col=rgb(0.75,0.75,0.75,0.5), breaks = 50,
      xlab = 'Prevalence', xlim = (c(0, 25)),
      ylab = 'Probability Density',
      ylim = (c(0, 0.6)),
      main = 'Visualize Population & Samples')

# Overlay curve:
# Population (Prevalence) density
lines(density(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)

# Overlay line:
# mean = mean of Population (Prevalence)
abline(v=mean(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)

# Overlay:
# First few sample's density & mean
lines(density(clt_sample_k[, 1]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, 1]), col="blue", lwd=1)

lines(density(clt_sample_k[, 2]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, 2]), col="blue", lwd=1)

lines(density(clt_sample_k[, 3]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, 3]), col="blue", lwd=1)

lines(density(clt_sample_k[, 4]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, 4]), col="blue", lwd=1)

lines(density(clt_sample_k[, 5]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, 5]), col="blue", lwd=1)

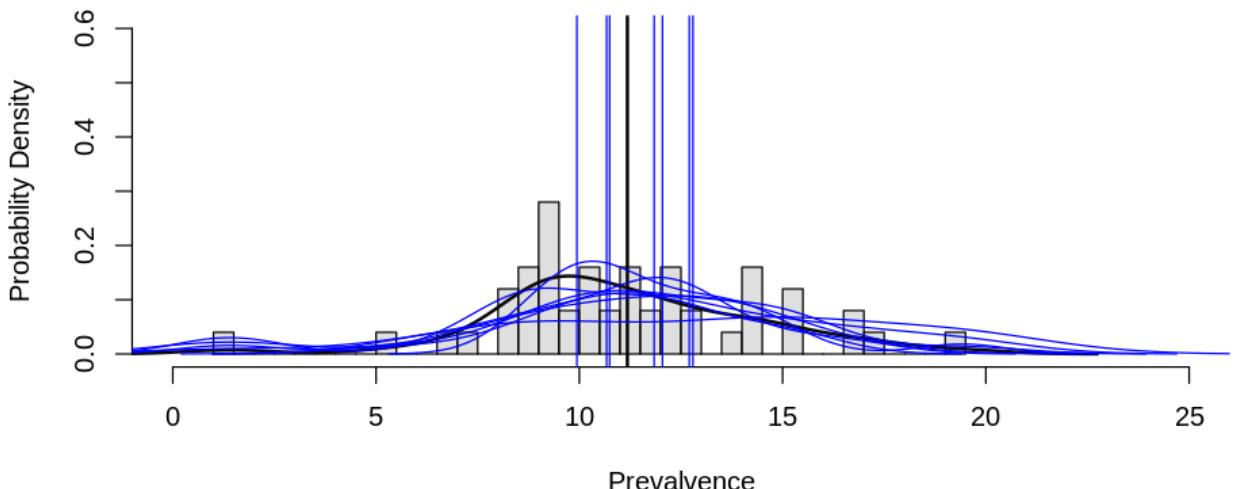
lines(density(clt_sample_k[, 6]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, 6]), col="blue", lwd=1)

lines(density(clt_sample_k[, clt_k]), col="blue", lwd=1)
abline(v=mean(clt_sample_k[, clt_k]), col="blue", lwd=1)

# We can see that sample's distributions are all different.

```

Visualize Population & Samples



[Tips] We notice that sample's distributions are all different.

Create Sampling Distribution (only one):

In [296]:

```
# -----  
# Sampling distribution (only one)  
# -----  
# Calculate sample mean value for k samples  
clt_sample_k_mean <- apply(clt_sample_k, 2, mean)  
# Show first few sample means  
clt_sample_k_mean[1:6]
```

10.745 9.94 12.705 10.67 12.045 12.795

In [297]:

```
# Calculate sample std-dev value for each individual sample (totally k std-dev  
clt_sample_k_sd <- apply(clt_sample_k, 2, sd)  
# Show first few samples' std-dev  
clt_sample_k_sd[1:6]
```

3.52158382430107 3.7529497170822 3.57115269611191 4.01026314926038
3.17895630009202 5.01140541270873

In [298]:

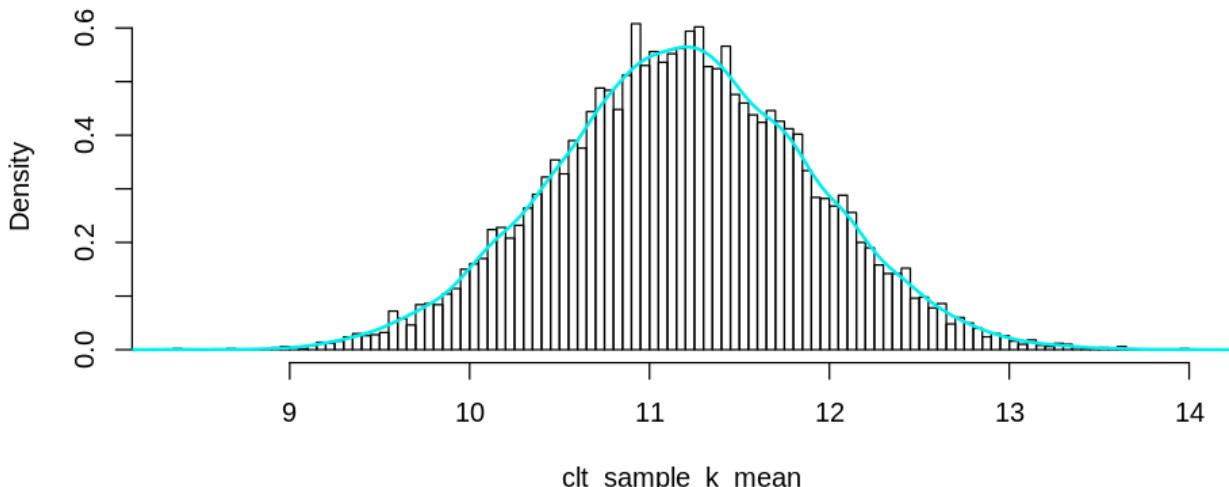
```
# Calculate std-dev value for Sampling DIstdtribution (only one std-dev)  
sd(clt_sample_k_mean)
```

0.713336703117785

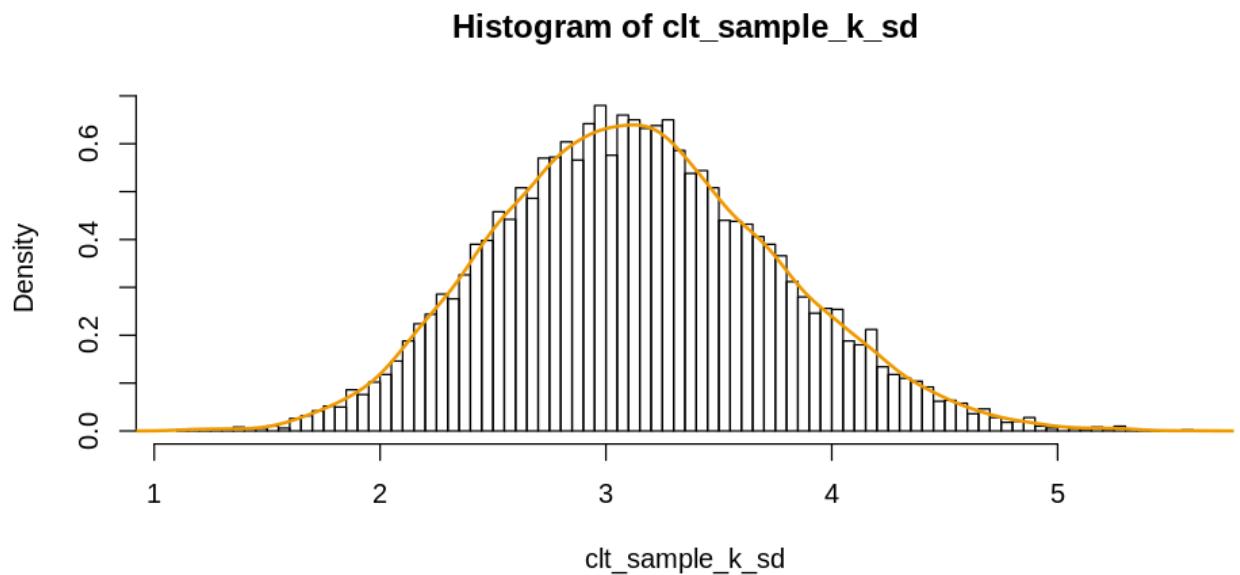
In [299]:

```
# histogram of sample means (Sampling distribution of the mean)  
hist(clt_sample_k_mean, probability = T, breaks = 100)  
lines(density(clt_sample_k_mean), col="cyan2", lwd=2)
```

Histogram of clt_sample_k_mean



```
In [300]: # histogram of sample std-dev  
hist(clt_sample_k_sd, probability = T, breaks = 100)  
lines(density(clt_sample_k_sd), col="orange2", lwd=2)
```



```
In [301]: # k *Sample* (sample size = n) mean Prevalence  
mean(clt_sample_k_mean)
```

11.1788145

```
In [302]: # *Population* mean Prevalence  
mean(ASD_State_SPED_2016$Prevalence)
```

11.182

[Tips] We notice that the above two means are close.

Visualise: Central Limit Theorem (CLT)

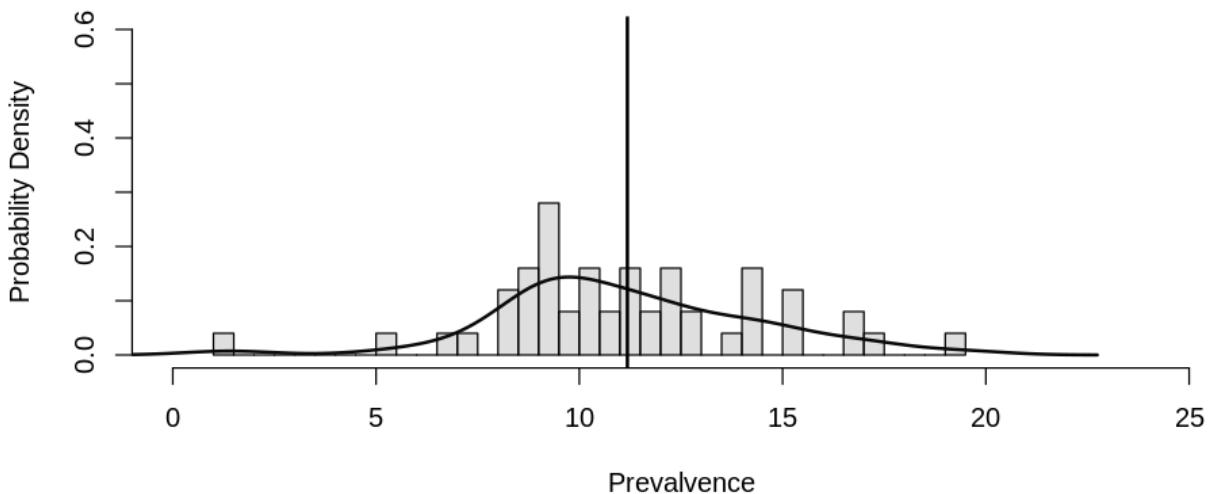
In [303]:

```
# -----
# Sampling distribution vs. Population distribution vs. Z-Norm
# -----
# Create:
# Population (Prevalence) histogram in probability
hist(ASD_State_SPED_2016$Prevalence, probability = T,
      col=rgb(0.75,0.75,0.75,0.5), breaks = 50,
      xlab = 'Prevalvence', xlim = (c(0, 25)),
      ylab = 'Probability Density', ylim = (c(0, 0.6)),
      main = 'Visualize Central Limit Theorem (CLT)')

# Overlay curve:
# Population (Prevalence) density
lines(density(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)

# Overlay line:
# mean = mean of Population (Prevalence)
abline(v=mean(ASD_State_SPED_2016$Prevalence), col="black", lwd=2)
```

Visualize Central Limit Theorem (CLT)

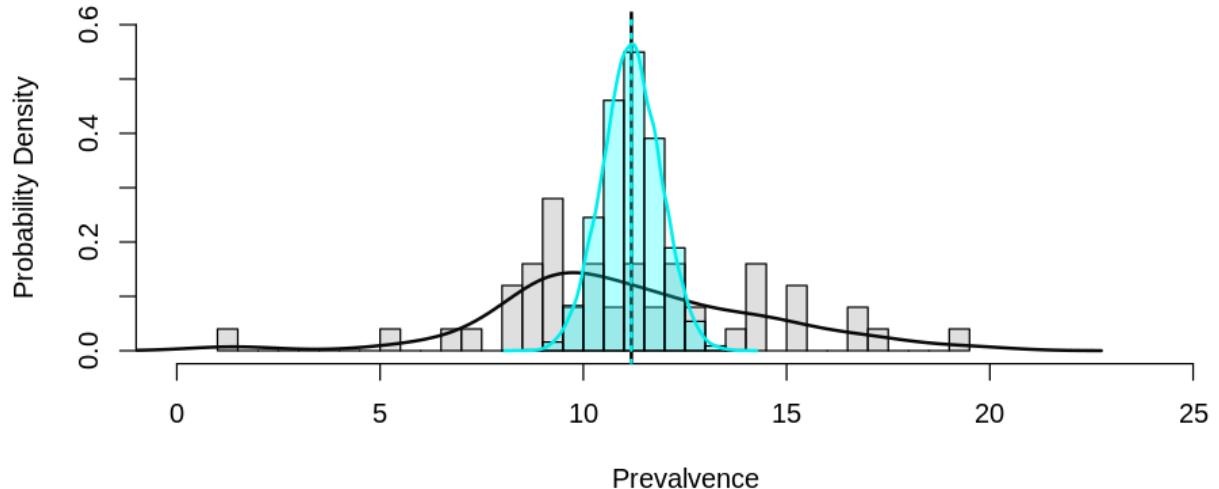


Overlay: Sample means histogram in probability (Sampling disribution)

In [304]:

```
# -----  
# Sampling distribution vs. Population distribution vs. Z-Norm  
# -----  
# Create:  
# Population (Prevalence) histogram in probability  
hist(ASD_State_SPED_2016$Prevalence, probability = T,  
    col=rgb(0.75,0.75,0.75,0.5), breaks = 50,  
    xlab = 'Prevalence', xlim = (c(0, 25)),  
    ylab = 'Probability Density', ylim = (c(0, 0.6)),  
    main = 'Visualize Central Limit Theorem (CLT)')  
  
# Overlay curve:  
# Population (Prevalence) density  
lines(density(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)  
  
# Overlay line:  
# mean = mean of Population (Prevalence)  
abline(v=mean(ASD_State_SPED_2016$Prevalence), col="black", lwd=2)  
  
# Overlay line:  
# Sample means histogram in probability (Sampling distribution)  
hist(clt_sample_k_mean, probability = T,  
    col=rgb(0,1,1,0.3), # https://www.dataanalytics.org.uk/make-transparent-c  
    add=T)  
  
# Overlay curve:  
# Sample (Prevalence) density (Sampling distribution)  
lines(density(clt_sample_k_mean), col="cyan2", lwd=2)  
# Overlay line:  
# mean of Sampling distribution (of Prevalence, sample size n)  
abline(v=mean(clt_sample_k_mean), col="cyan2", lwd=2, lty=3)
```

Visualize Central Limit Theorem (CLT)



< How to make transparent colors in R >

<https://www.dataanalytics.org.uk/make-transparent-colors-in-r/> (<https://www.dataanalytics.org.uk/make-transparent-colors-in-r/>)

col2rgb(c("cyan", "grey", "red")) / 255

```
In [305]: col2rgb(c("cyan", "grey", "purple", "orange")) / 255
```

```
red  0  0.745098  0.6274510  1.0000000
green 1  0.745098  0.1254902  0.6470588
blue 1  0.745098  0.9411765  0.0000000
```

```
In [306]: # Recall:
```

```
# k *Sample* (sample size = n) mean Prevalence
mean(clt_sample_k_mean)
# *Population* mean Prevalence
mean(ASD_State_SPED_2016$Prevalence)
# We see that the above two means are close. Good estimation!
```

```
11.1788145
```

```
11.182
```

[Tips] We notice that the above two means are close. Good estimation!

Standard Error (SE) (of mean prevalence), can be estimated as: std-dev of the Sampling distribution (of mean prevalence):

```
In [307]:
```

```
# -----
# Standard Error (SE) (of mean prevalence), can be estimated as:
# std-dev of the Sampling distribution (of mean prevalence)
# -----
# https://en.wikipedia.org/wiki/Sampling_distribution

# [1] Actual SE: When Population std-dev is known, SE using Population standard deviation
sd.p(ASD_State_SPED_2016$Prevalence) / sqrt(clt_n)

# [2] Estimated SE with k samples: When Population std-dev is NOT known, but we have k samples
sd(clt_sample_k_mean)

# [3] Estimated SE with only one sample: When Population std-dev is NOT known, but we have only one sample
clt_sample_k_sd[1] / sqrt(clt_n)
```

```
0.715705106870141
```

```
0.713336703117785
```

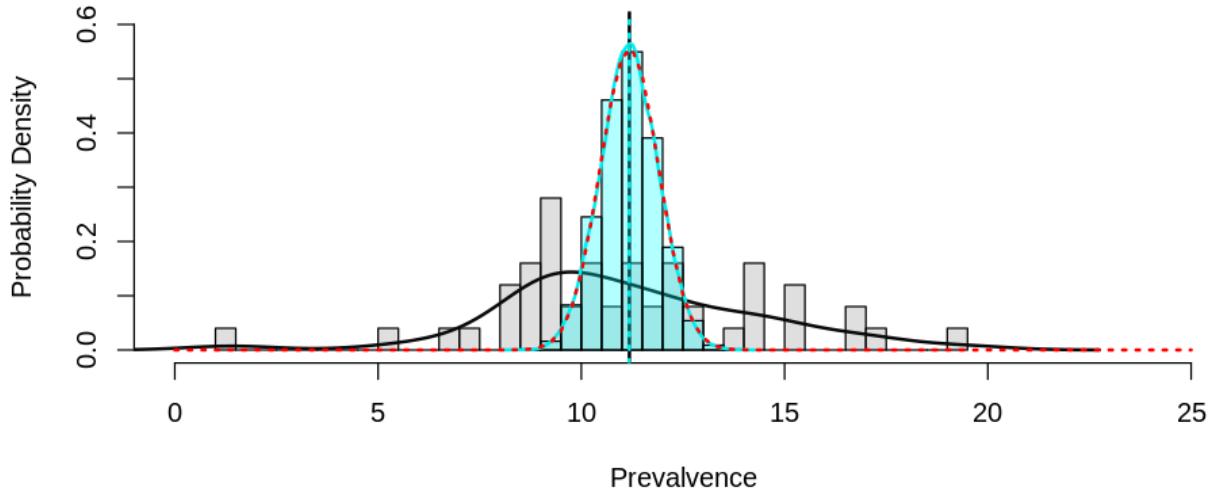
```
0.787450081960087
```

Overlay: **Theoretic Sampling Distribution** with population mean & std-dev = Actual SE:

In [308]:

```
# -----  
# Sampling distribution vs. Population distribution vs. Z-Norm  
# -----  
# Create:  
# Population (Prevalence) histogram in probability  
hist(ASD_State_SPED_2016$Prevalence, probability = T,  
    col=rgb(0.75,0.75,0.75,0.5), breaks = 50,  
    xlab = 'Prevalvence', xlim = (c(0, 25)),  
    ylab = 'Probability Density', ylim = (c(0, 0.6)),  
    main = 'Visualize Central Limit Theorem (CLT)')  
  
# Overlay curve:  
# Population (Prevalence) density  
lines(density(ASD_State_SPED_2016$Prevalence), col="grey4", lwd=2)  
  
# Overlay line:  
# mean = mean of Population (Prevalence)  
abline(v=mean(ASD_State_SPED_2016$Prevalence), col="black", lwd=2)  
  
# Overlay line:  
# Sample means histogram in probability (Sampling disribution)  
hist(clt_sample_k_mean, probability = T,  
    col=rgb(0,1,1,0.3), # https://www.dataanalytics.org.uk/make-transparent-c  
    add=T)  
  
# Overlay curve:  
# Sample (Prevalence) density (Sampling disribution)  
lines(density(clt_sample_k_mean), col="cyan2", lwd=2)  
# Overlay line:  
# mean of Sampling distribution (of Prevelance, sample size n)  
abline(v=mean(clt_sample_k_mean), col="cyan2", lwd=2, lty=3)  
  
# Overlay curve:  
# *Theoretic Sampling Distribution* with population mean & std-dev = Actual SE  
# mean = mean of Population (Prevalence) & std-dev = std-dev of Population (Pr  
curve(dnorm(x,  
    mean(ASD_State_SPED_2016$Prevalence), # Actual Population mean  
    sd.p(ASD_State_SPED_2016$Prevalence) / sqrt(clt_n)), # Actual SE (Pr  
    add=TRUE, col="red", lwd=2, lty=3)
```

Visualize Central Limit Theorem (CLT)

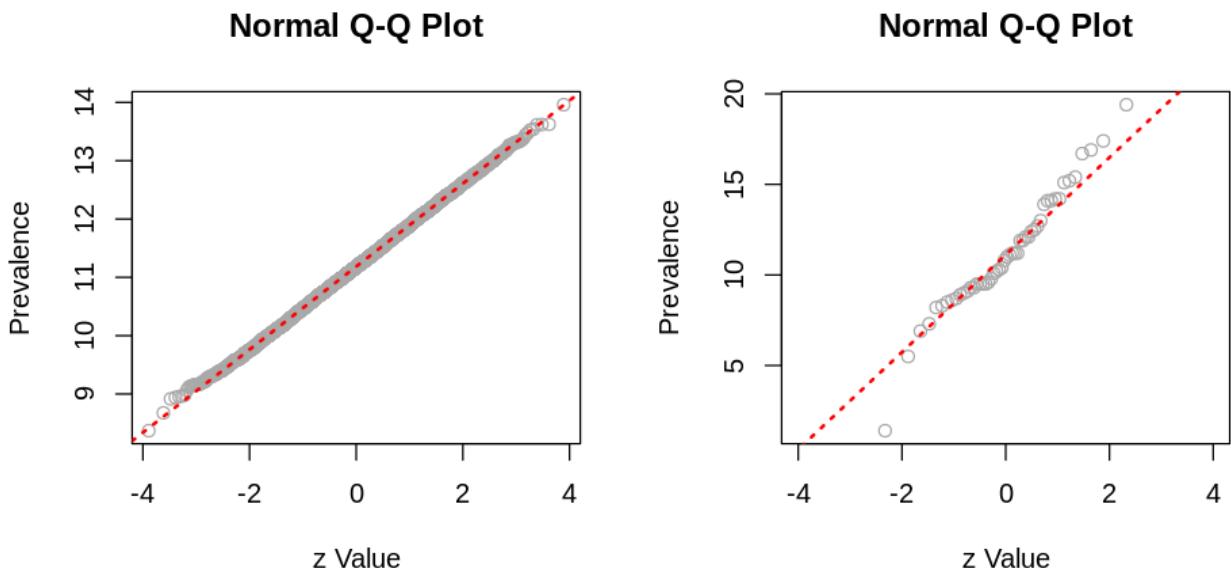


Use QQ Plot

< Construct a Quantile-Quantile Plot (QQ plot) > <https://youtu.be/okjYjCISj0g>
[\(https://youtu.be/okjYjCISj0g\)](https://youtu.be/okjYjCISj0g)

In [309]:

```
# -----  
# Evaluate normality  
# -----  
# Construct a Quantile-Quantile Plot (QQ plot)  
# https://youtu.be/okjYjCISj0g  
  
par(mfrow=c(1, 2))  
# Sample means  
qqnorm(clt_sample_k_mean, col="darkgrey",  
#       xlim=(c(-4, 4)), ylim=(c(0, 20)),  
#       xlab="z Value", ylab="Prevalence")  
qqline(clt_sample_k_mean, col="red", lwd=2, lty=3)  
# Population  
qqnorm(ASD_State_SPED_2016$Prevalence, col="darkgrey",  
#       xlim=(c(-4, 4)),  
#       xlab="z Value", ylab="Prevalence")  
qqline(ASD_State_SPED_2016$Prevalence, col="red", lwd=2, lty=3)  
# Reset  
par(mfrow=c(1, 1))
```

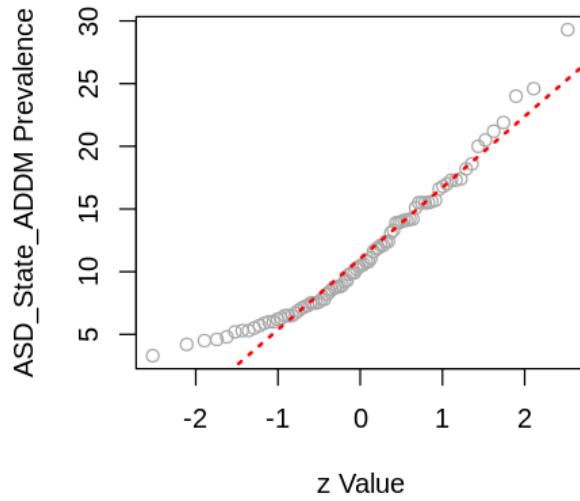


[Tips] If most/all data points are aligned with the red straight line, then the underlying data points are normally distributed.

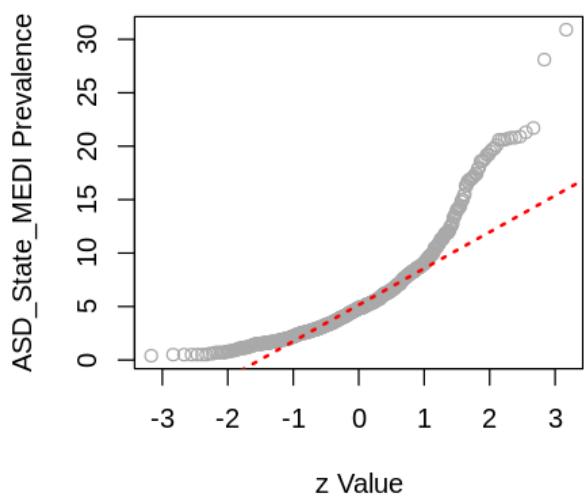
In [310]:

```
# -----  
# Evaluate normality  
# -----  
par(mfrow=c(1, 2))  
  
# ASD_State_ADDM$Prevalence  
qqnorm(ASD_State_ADDM$Prevalence, col="darkgrey",  
#       xlim=(c(-4, 4)), ylim=(c(0, 20)),  
#       xlab="z Value", ylab="ASD_State_ADDM Prevalence")  
qqline(ASD_State_ADDM$Prevalence, col="red", lwd=2, lty=3)  
# plot(density(ASD_State_ADDM$Prevalence))  
  
# ASD_State_MEDI$Prevalence  
qqnorm(ASD_State_MEDI$Prevalence, col="darkgrey",  
#       xlim=(c(-4, 4)), ylim=(c(0, 20)),  
#       xlab="z Value", ylab="ASD_State_MEDI Prevalence")  
qqline(ASD_State_MEDI$Prevalence, col="red", lwd=2, lty=3)  
  
# ASD_State_NSCH$Prevalence  
qqnorm(ASD_State_NSCH$Prevalence, col="darkgrey",  
#       xlim=(c(-4, 4)), ylim=(c(0, 20)),  
#       xlab="z Value", ylab="ASD_State_NSCH Prevalence")  
qqline(ASD_State_NSCH$Prevalence, col="red", lwd=2, lty=3)  
  
# ASD_State_SPED$Prevalence  
qqnorm(ASD_State_SPED$Prevalence, col="darkgrey",  
#       xlim=(c(-4, 4)), ylim=(c(0, 20)),  
#       xlab="z Value", ylab="ASD_State_SPED Prevalence")  
qqline(ASD_State_SPED$Prevalence, col="red", lwd=2, lty=3)  
  
# Reset  
par(mfrow=c(1, 1))
```

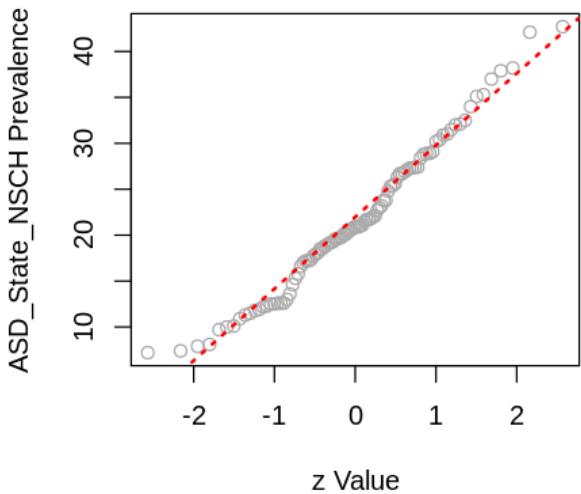
Normal Q-Q Plot



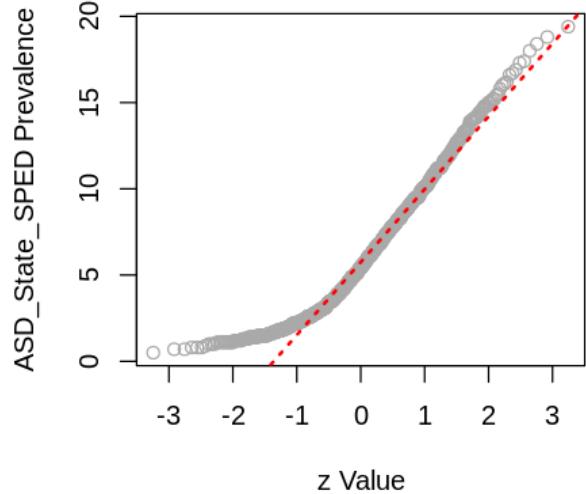
Normal Q-Q Plot



Normal Q-Q Plot



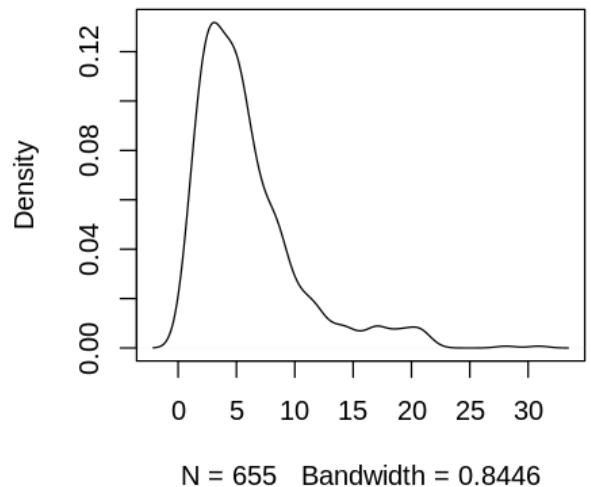
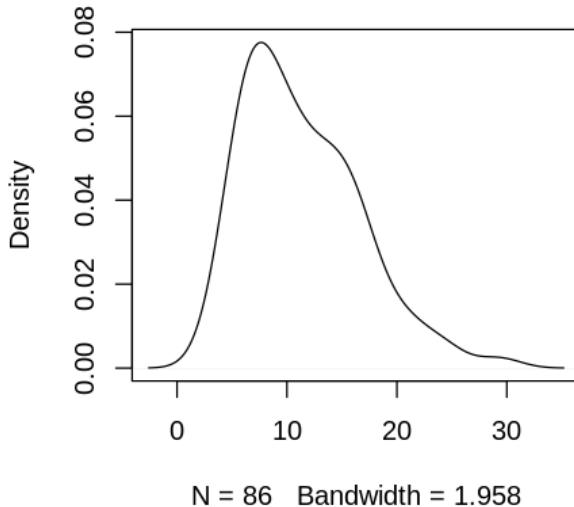
Normal Q-Q Plot



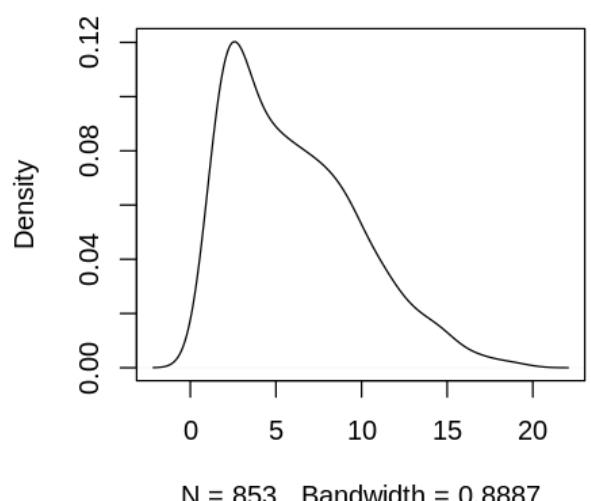
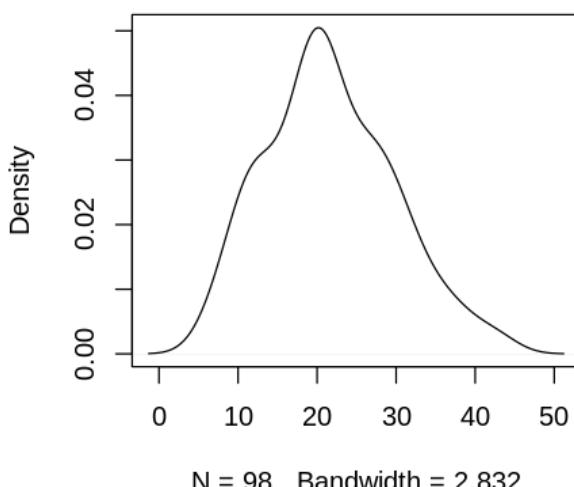
In [311]:

```
# -----  
# Evaluate normality  
# -----  
par(mfrow=c(1, 2))  
plot(density(ASD_State_ADDM$Prevalence))  
plot(density(ASD_State_MEDI$Prevalence))  
plot(density(ASD_State_NSCH$Prevalence))  
plot(density(ASD_State_SPED$Prevalence))  
# Reset  
par(mfrow=c(1, 1))
```

density.default(x = ASD_State_ADDM\$Prevalence) density.default(x = ASD_State_MEDI\$Prevalence)



density.default(x = ASD_State_NSCH\$Prevalence) density.default(x = ASD_State_SPED\$Prevalence)



```
In [312]: # Alternatively, use shapiro.test() to test Normality
set.seed(88)

# Test data of k sample's means (Sampling Distribution data):
shapiro.test(sample(x = clt_sample_k_mean, size = 1000))

# Test data of population's Prevalence values (Population Distribution data):
shapiro.test(ASD_State_SPED_2016$Prevalence)
```

```
Shapiro-Wilk normality test

data: sample(x = clt_sample_k_mean, size = 1000)
W = 0.99904, p-value = 0.8907
```

```
Shapiro-Wilk normality test

data: ASD_State_SPED_2016$Prevalence
W = 0.96985, p-value = 0.2282
```

[Tips] General speaking, if **p-value** is greater than **0.05** (meaning more than 5% chance of being normally distributed), then the underlying data points are normally distributed.

Confidence Interval (CI)

Confidence Interval (CI) - Mean Estimation & Its CI

Use a **Sample statistic (e.g. mean)** to estimate a **population statistic (e.g. mean)**. And quantitatively calculate the confidence of the estimation.

In [313]:

```
# -----  
# Use a sample of a few US. State's ASD prevalence (mean) to estimate:  
# Average prevalence of ALL US. States (the *Population*) [Source SPED, Year 2  
# -----  
dim(ASD_State_SPED_2016)  
#  
ASD_State_SPED_2016 # This is considered as a population now.
```

50 2

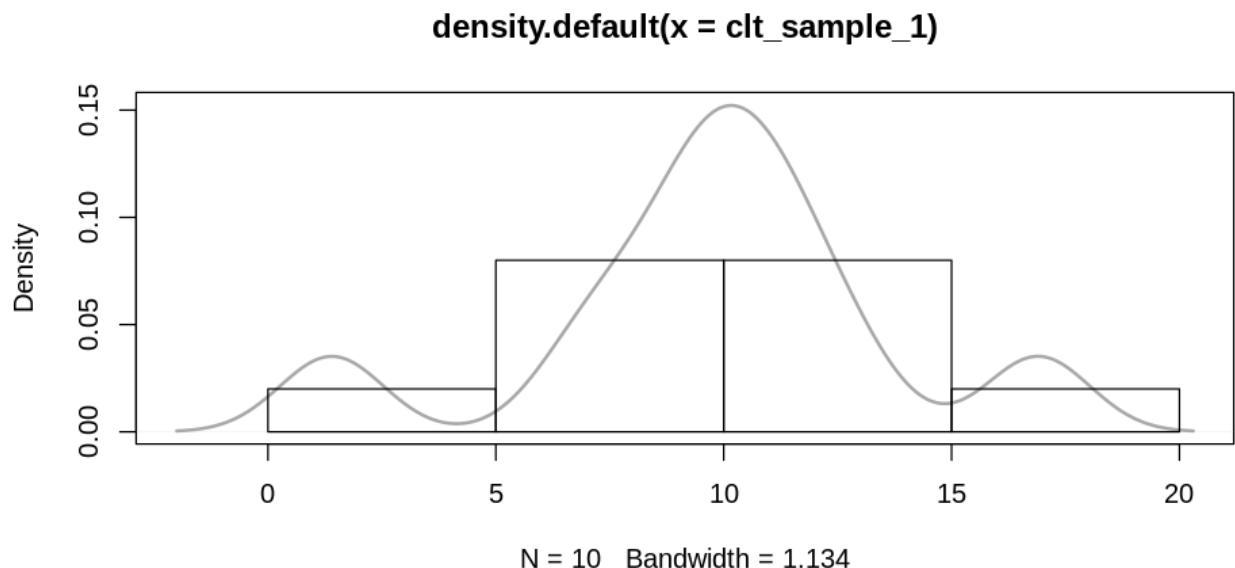
Draw a **Sample** from ASD_State_SPED_2016\$Prevalence

In [314]:

```
# Create a *Sample* from ASD_State_SPED_2016$Prevalence,  
# with sample size n =  
clt_n = 10  
# Try 20 or 40, larger sample size, narrower the CI (more confident at xx% lev  
# clt_n = 20  
  
set.seed(88)  
clt_sample_1 = sample(x = ASD_State_SPED_2016$Prevalence, size = clt_n, replace = TRUE)
```

11.2 9.5 16.9 6.9 11.2 8.2 12.7 9.5 1.4 10.2

```
In [315]: plot(density(clt_sample_1), col="darkgrey", lwd=2)
hist(clt_sample_1, probability = T, add = T)
```



```
In [316]: # Sample mean Prevalence
mean(clt_sample_1)

# *Population* mean Prevalence
mean(ASD_State_SPED_2016$Prevalence)
```

9.77

11.182

1. Calculate Confidence Interval of mean estimation: CI using Z (Standard Normal) distribution

```
In [317]: # -----
# CI using Z (Standard Normal) distribution
# -----
# sample mean
sample_mean = mean(clt_sample_1)
sample_mean
```

9.77

```
In [318]: # sample size n
sample_size_n = length(clt_sample_1)
sample_size_n
```

10

```
In [319]: # sample standard deviation
sample_sd = sd(clt_sample_1)
sample_sd
```

4.00833853083516

```
In [320]: # sample standard error
sample_se = sample_sd / sqrt(sample_size_n)
sample_se
```

1.26754793904522

```
In [321]: # 95% quantile (z score)
z_score = qnorm(p = 0.975)
z_score
```

1.95996398454005

```
In [322]: # ?qnorm
```

```
In [323]: # CI using Z distribution
sample_ci = z_score * sample_se
sample_ci
```

2.4843483092066

```
In [324]: # Lower CI: mean + CI
sample_mean - sample_ci
```

```
# Upper CI: mean + CI
sample_mean + sample_ci
```

```
# Display
```

```
cat('\t< Confidence Interval (Prevalence) >\n', '\tLower CI : ', sample_mean)
```

7.2856516907934

12.2543483092066

```
< Confidence Interval (Prevalence) >
Lower CI : 7.285652    Mean : 9.77    Upper CI : 12.25435
```

[?] Is the population mean in this CI range?

2. Calculate Confidence Interval of mean estimation: CI using T distribution

```
In [325]: # -----
# CI using T distribution
# -----
# sample mean
sample_mean = mean(clt_sample_1)
sample_mean
# sample size n
sample_size_n = length(clt_sample_1)
sample_size_n
# sample standard deviation
sample_sd = sd(clt_sample_1)
sample_sd
# sample standard error
sample_se = sample_sd / sqrt(sample_size_n)
sample_se
```

9.77

10

4.00833853083516

1.26754793904522

```
In [326]: # 95% quantile (t score)
t_score = qt(p = 0.975, df = sample_size_n - 1)
t_score
```

2.2621571627982

```
In [327]: # ?qt
```

```
In [328]: # CI using T distribution
sample_ci = t_score * sample_se
sample_ci

# Lower CI: mean + CI
sample_mean - sample_ci

# Upper CI: mean + CI
sample_mean + sample_ci

# Display
cat('\t< Confidence Interval (Prevalence) >\n', '\tLower CI : ', sample_mean
```

2.86739264950124

6.90260735049876

12.6373926495012

< Confidence Interval (Prevalence) >

Lower CI : 6.902607 Mean : 9.77 Upper CI : 12.63739

[?] Is the population mean in this CI range?

[?] Compare CIs of Z and T distribution, which CI has wider range? Is it reasonable?

```
In [329]: # Alternatively, calculate CI using t.test() function
t.test(clt_sample_1, conf.level = 0.95)
```

One Sample t-test

```
data: clt_sample_1
t = 7.7078, df = 9, p-value = 2.976e-05
alternative hypothesis: true mean is not equal to 0
95 percent confidence interval:
 6.902607 12.637393
sample estimates:
mean of x
 9.77
```

```
In [330]: # Two group hypothesis test : sample mean vs. population mean  
t.test(clt_sample_1, conf.level = 0.95, mu = mean(ASD_State_SPED_2016$Prevalen
```

One Sample t-test

```
data: clt_sample_1  
t = -1.114, df = 9, p-value = 0.2942  
alternative hypothesis: true mean is not equal to 11.182  
95 percent confidence interval:  
 6.902607 12.637393  
sample estimates:  
mean of x  
 9.77
```

Quiz:

Obtain CI using smaller/larger sample size (clt_n) at 99% confidence. Compare CI width.
Observe: larger sample size, narrower the CI (more confident at xx% level)

```
In [331]: # Write your code below and press Shift+Enter to execute
```

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

Confidence Interval (CI) - Proportion Estimation & Its CI

Use a **Sample statistic (e.g. proportion)** to estimate a **population statistic (e.g. proportion)**. And quantitatively calculate the confidence of the estimation.

```
In [332]: # -----  
# Use a sample of one US. State's ASD prevalence (proportion) to estimate:  
# Prevalence of THAT US. State's ALL Children (the *Population*) [Source SPED,  
# -----  
  
# No. Children with ASD  
ASD <- ASD_State_SPED$Numerator_ASD[ASD_State_SPED$Year == 2016]  
#  
str(ASD)
```

```
int [1:50] 6140 1204 10746 4181 79041 5902 7391 1383 782 30920 ...
```

```
In [333]: # No. Children with ASD of first US. State (AL-Alabama)  
ASD[1]
```

6140

```
In [334]: # No. Children surveyed
Children <- ASD_State_SPED$Denominator[ASD_State_SPED$Year == 2016]
#
str(Children)

  int [1:50] 674701 119217 1033241 440130 5686400 808556 479961 124609 65732 2
  555399 ...
```

```
In [335]: # No. Children surveyed of first US. State (AL-Alabama)
Children[1]
```

674701

1. Calculate Confidence Interval of proportion estimation: CI using Z score interval (standard normal distribution)

```
In [336]: # -----
# CI using Z score interval (standard normal distribution)
# https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval
# -----
```

```
# sample proportion of first US. State (AL-Alabama) in year 2016 of source SPE
sample_proportion = ASD[1] / Children[1]
sample_proportion
```

0.00910032740428723

```
In [337]: # sample size n
sample_size_n = Children[1]
sample_size_n
```

674701

```
In [338]: # 95% quantile (z score)
z_score = qnorm(p = 0.975)
z_score
```

1.95996398454005

```
In [339]: sample_ci = z_score * sqrt(sample_proportion * (1 - sample_proportion)) / sample_size_n
sample_ci
```

0.000226587404757579

```
In [340]: # Lower CI: mean + CI
sample_proportion - sample_ci

# Upper CI: mean + CI
sample_proportion + sample_ci

# Display
cat('\t< Confidence Interval >\n', '\tLower CI : ', sample_proportion - sample_ci, '\n', '\tUpper CI : ', sample_proportion + sample_ci, '\n')
```

0.00887373999952965

0.00932691480904481

< Confidence Interval >

Lower CI : 0.00887374 Mean : 0.009100327 Upper CI : 0.00932691480904481

```
In [341]: # Display * 1000 -> Prevalence
cat('\t< Confidence Interval (Prevalence) >\n', '\tLower CI : ', 1000*(sample[1]*1000))

< Confidence Interval (Prevalence) >
Lower CI : 8.87374      Mean : 9.100327      Upper CI : 9.326915
```

[Tips] Based above calculation upon [Source: SPED] [Year: 2016] data, we have 95% confidence that: The actual AL-Alabama state level ASD prevalence (if ALL childrens in Alabama state were surveyed) would be in the above calculated CI range 95% times.

Or, asumming there are 100 independently different Alabama states exist in 100 parallel universe, we obtained 100 actual prevelance proportions. 95 of them will likely fall into the CI value range.

2. Calculate Confidence Interval of proportion estimation: CI using Wilson score interval

```
In [342]: # -----
# CI using Wilson score interval
# https://en.wikipedia.org/wiki/Binomial_proportion_confidence_interval
# -----
```

```
sample_proportion # = ASD[1] / Children[1]

# Yates' chi-squared test = Wilson score interval with continuity correction -
prop.test(ASD[1], Children[1], conf.level = 0.95)

# Pearson's chi-squared test = Wilson score interval - wilson
prop.test(ASD[1], Children[1], conf.level = 0.95, correct = FALSE)
```

0.00910032740428723

1-sample proportions test with continuity correction

```
data: ASD[1] out of Children[1], null probability 0.5
X-squared = 650363, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.008875786 0.009330477
sample estimates:
      p
0.009100327
```

1-sample proportions test without continuity correction

```
data: ASD[1] out of Children[1], null probability 0.5
X-squared = 650365, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.008876518 0.009329726
sample estimates:
      p
0.009100327
```

Quiz:

Obtain CI of Male.Prevalence propotion [Source: ADDM] [Year: 2014] at 99% confidence.

In [343]: # Write your code below and press Shift+Enter to execute

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

Quiz:

Obtain CI of Female.Prevalence propotion [Source: ADDM] [Year: 2014] at 99% confidence.

Then Compare CI range with Male children's CI range. Which gender has statistically higer ASD prevalence/proportion?

In [344]: # Write your code below and press Shift+Enter to execute

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

.0

Workshop Submission

What to submit?

Choose one of below visualisations/charts, use R to construct the chart nicely.

Optionally, enhance it with additional data dimensions to be better than original chart.

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

Select data source: ADDM Network*

ASD Data Collection Locations for:
ADDM Network*

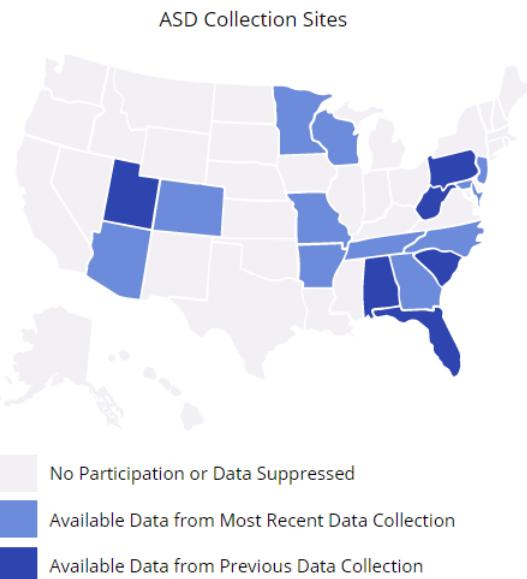
Since the launch of the ADDM Network in 2000, CDC has funded **16 sites** at various times. In 2014, ASD data were collected from **11 sites** by obtaining the health and education records of children with behaviors consistent with ASD.

WHY THIS MATTERS

When reviewing ASD data and findings, it is important to consider *where* the data were collected and *how* each location might affect the data. Across the United States, each community has a unique population with different characteristics. There are also regional differences in healthcare and education systems, which can affect *when* and *how* children with ASD are identified, as well as the services they receive.

Because of these geographic differences, it may not be possible to directly compare data collected in one community to data from other communities. Take ADDM Network data collected from 11 sites in 2014, for example. In Colorado, ASD prevalence was 13.9 out of 1,000 kids, whereas in North Carolina, ASD prevalence was 17.4 out of 1,000 kids. There is a clear difference in the number of children identified with ASD in these two states, but without additional information, it is difficult to know *why* these differences exist. Therefore, it would not be correct to assume that the prevalence in one state will be the same as another state.

*ADDM data do not represent the entire state, only a selection of sites within the state.



2014 ADDM NETWORK DATA

In this section, explore the most recent ADDM data, both overall and among certain demographic groups by study area.

Select a location: U.S. or Total†

MOST RECENT STUDY YEAR: 2014

ASD PREVALENCE PER 1,000 8-YEAR-OLD CHILDREN

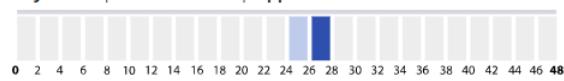
Prevalence Overall

Overall: 16.8 | Lower CI: 16.4 | Upper CI: 17.3

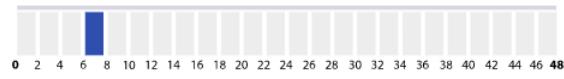


Prevalence By Sex

Boys: 26.6 | Lower CI: 25.8 | Upper CI: 27.4



Girls: 6.6 | Lower CI: 6.2 | Upper CI: 7

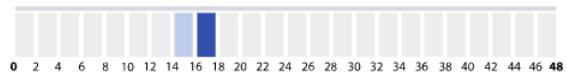


Prevalence By Race/Ethnicity

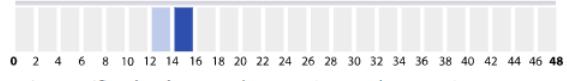
Non-Hispanic White: 17.2 | Lower CI: 16.5 | Upper CI: 17.8



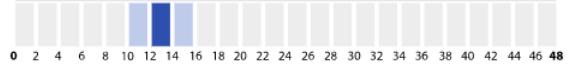
Non-Hispanic Black: 16 | Lower CI: 15.1 | Upper CI: 16.9



Hispanic: 14 | Lower CI: 13.1 | Upper CI: 14.9



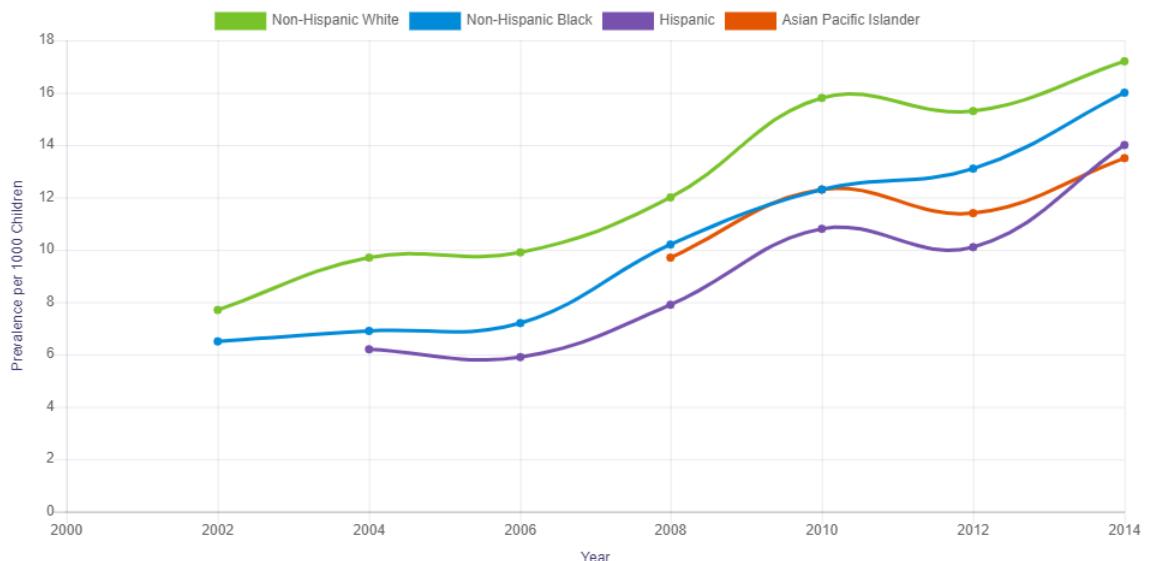
Asian/Pacific Islander: 13.5 | Lower CI: 11.8 | Upper CI: 15.4



†ADDM estimate = the total for all sites combined.

Prevalence Estimates by Race/Ethnicity

Show ADDM prevalence estimates* by race/ethnicity for: ▾



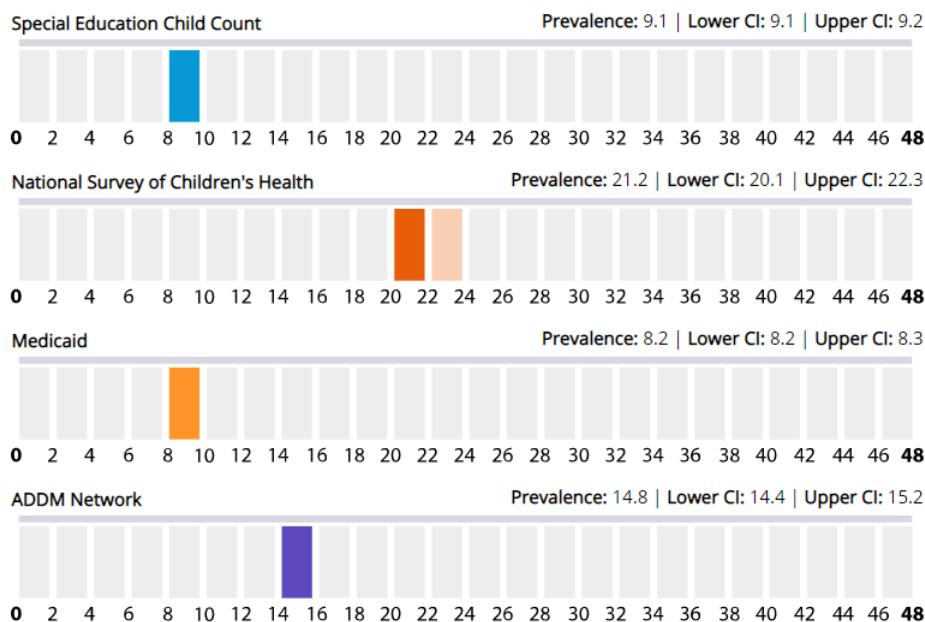
Note: Click the icons and racial/ethnic groups above the chart to hide or unhide data. Hover your mouse over data points to show prevalence by year.

*ADDM data do not represent the entire state, only a selection of sites within the state.

[†]ADDM estimate = the total for all sites combined.

Confidence Intervals by Data Set/Location

Select state: ▾



WHY THIS MATTERS

By comparing different data sets, we see that some confidence intervals are wide, while others are narrow. When a confidence interval is wide, the true prevalence may be anywhere within that range, making it less certain. A narrow confidence interval means we can be more certain about the reported prevalence.

Note: The graph above shows data from 2012, the most recent year for which all data sets had data.

[†]ADDM estimate = the total for all sites combined.

In [345]: # Write your code below and press Shift+Enter to execute

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.

0

Appendices

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

In [346]:

```
# -----
# Interactive workshops: < Learning R inside R > using swirl() (in R/RStudio)
# -----  
  
# https://github.com/telescopeuser/S-SB-Workshop  
  
# Restart R to install to install "swirl" package:  
  
# if(!require(swirl)){install.packages("swirl")}  
# library("swirl")  
# install_course("R Programming")  
# install_course("Exploratory Data Analysis")  
# install_course("Regression Models")  
# install_course("Statistical Inference")  
# swirl()
```

Neural Network 101 using nnet()

In [347]:

```
# -----
# Neural Network 101 using nnet()
# -----
if(!require(nnet)){install.packages("nnet")}
library("nnet")
# ?nnet

# < Case: predict three different iris flower types >

# https://en.wikipedia.org/wiki/Iris_flower_data_set
# https://archive.ics.uci.edu/ml/datasets/iris

# Data preparation: split iris data in two halves, for training & testing resp
ir <- rbind(iris3[,1],iris3[,2],iris3[,3])
targets <- class.ind( c(rep("s", 50), rep("c", 50), rep("v", 50)) )
samp <- c(sample(1:50,25), sample(51:100,25), sample(101:150,25))
# Model training (machine learning / data fitting)
irl <- nnet(ir[samp,], targets[samp,], size = 2, rang = 0.1,
            decay = 5e-4, maxit = 200)
# Model prediction
test.cl <- function(true, pred) {
  true <- max.col(true)
  cres <- max.col(pred)
  table(true, cres)
}

# Model evaluation
test.cl(targets[-samp,], predict(irl, ir[-samp,]))
```

Loading required package: nnet

```
# weights:  19
initial value 56.720484
iter  10 value 48.312067
iter  20 value 29.096504
iter  30 value 18.239805
iter  40 value 17.345523
iter  50 value 17.180673
iter  60 value 16.944589
iter  70 value 16.897683
iter  80 value 16.883344
iter  90 value 16.882703
final  value 16.882577
converged

      cres
true  1  2  3
  1 20  0  5
  2  0 25  0
  3  0  0 25
```

Hypothesis Test - Mean - Z Test & t.test()

In [348]:

```
# -----  
# Hypothesis Test - Mean - Z Test & t.test()  
# -----  
  
# Create sample 1  
set.seed(88)  
sample_1 = rnorm(n = 10)  
# sample_1 = rnorm(n = 1000)  
  
# Create sample 2  
set.seed(88)  
sample_2 = rchisq(n = 10, df = 30) - 29  
# sample_2 = rchisq(n = 1000, df = 30) - 29  
  
mean(sample_1)  
mean(sample_2)  
  
par(mfrow=c(1, 2))  
hist(sample_1)  
hist(sample_2)  
par(mfrow=c(1, 1))  
  
t.test(sample_1, sample_2)
```

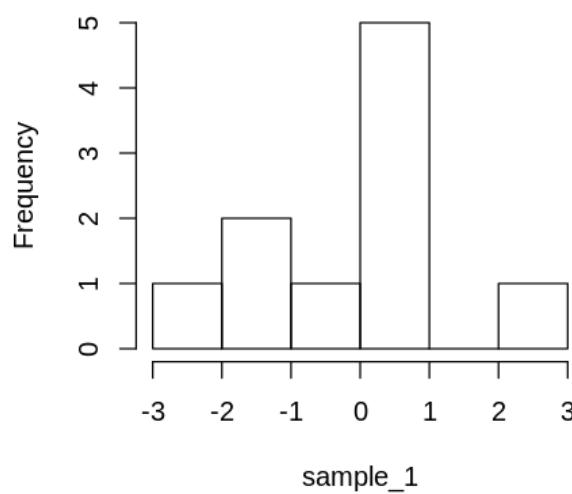
-0.10035252728447

1.47209538362297

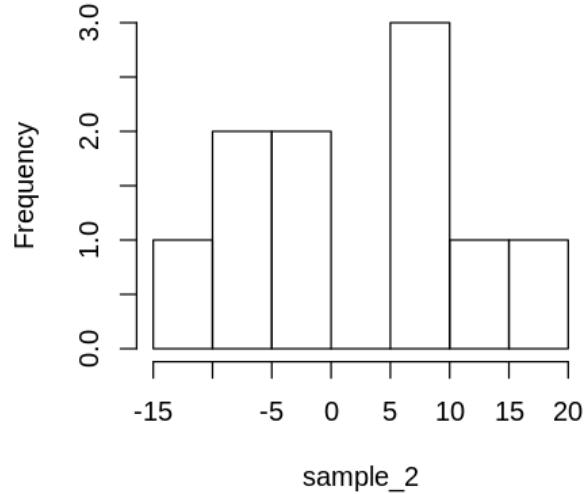
Welch Two Sample t-test

```
data: sample_1 and sample_2  
t = -0.53406, df = 9.3705, p-value = 0.6057  
alternative hypothesis: true difference in means is not equal to 0  
95 percent confidence interval:  
 -8.193020  5.048124  
sample estimates:  
 mean of x  mean of y  
-0.1003525  1.4720954
```

Histogram of sample_1



Histogram of sample_2



In [349]:

```
# -----  
# Hypothesis Test - Proportion - prop.test()  
# -----  
if(!require(binom)){install.packages("binom")}  
library('binom')  
  
# Different flavours of proportion test:  
binom.confint (x=ASD[1], n=Children[1], conf.level =0.95, method="all")  
  
# Multiple group hypothesis test : proportions (Prevalence) among all US. Sta  
prop.test(ASD, Children)
```

Loading required package: binom

Warning message in binom.bayes(x, n, conf.level = conf.level, ...):

"1 confidence interval failed to converge (marked by '*').

Try changing 'tol' to a different value."

method	x	n	mean	lower	upper
agresti-coull	6140	674701	0.009100327	0.008876501	0.009329744
asymptotic	6140	674701	0.009100327	0.008873740	0.009326915
bayes*	6140	674701	0.009101055	0.008874879	0.009328049
cloglog	6140	674701	0.009100327	0.008875951	0.009329145
exact	6140	674701	0.009100327	0.008875108	0.009329776
logit	6140	674701	0.009100327	0.008876513	0.009329732
probit	6140	674701	0.009100327	0.008876195	0.009329400
profile	6140	674701	0.009100327	0.008875600	0.009328782
lrt	6140	674701	0.009100327	0.008845578	0.009330138
prop.test	6140	674701	0.009100327	0.008875786	0.009330477
wilson	6140	674701	0.009100327	0.008876518	0.009329726

50-sample test for equality of proportions without continuity correction

data: ASD out of Children

X-squared = 28593, df = 49, p-value < 2.2e-16

alternative hypothesis: two.sided

sample estimates:

prop 1	prop 2	prop 3	prop 4	prop 5	prop 6
0.009100327	0.010099231	0.010400284	0.009499466	0.013900007	0.007299433
prop 7	prop 8	prop 9	prop 10	prop 11	prop 12
0.015399168	0.011098717	0.011896793	0.012099872	0.010300278	0.008299863
prop 13	prop 14	prop 15	prop 16	prop 17	prop 18
0.009498530	0.011000237	0.014200361	0.001399357	0.008200786	0.009299486
prop 19	prop 20	prop 21	prop 22	prop 23	prop 24
0.006900015	0.016696937	0.011900547	0.017399589	0.011199751	0.019400428
prop 25	prop 26	prop 27	prop 28	prop 29	prop 30
0.009499812	0.012399961	0.005500221	0.010800730	0.012698882	0.014097784
prop 31	prop 32	prop 33	prop 34	prop 35	prop 36
0.014100222	0.008501570	0.012999924	0.011199812	0.009796534	0.012499686
prop 37	prop 38	prop 39	prop 40	prop 41	prop 42
0.008899480	0.015100037	0.016899883	0.015201438	0.009599935	0.008997355
prop 43	prop 44	prop 45	prop 46	prop 47	prop 48
0.009500416	0.010199940	0.008699269	0.012103956	0.014200156	0.011200227
prop 49	prop 50				
0.008601179	0.009299132				

Correlation of Numeric Variables

In [350]:

```
# -----
# Correlation of Numeric Variables
# -----
cor_df = select_if(ASD_State, is.numeric) # Select only numeric variables
cor_df = cor_df[, colSums(is.na(cor_df)) == 0] # Select variables without NA

# Compute correlation matrix for No-NA numeric variables:
cor_table = cor(cor_df)
cor_table
```

	Denominator	Prevalence	Lower.CI	Upper.CI	Year	Numerator
Denominator	1.00000000	-0.1374662	-0.07863304	-0.17389486	0.02851671	0.82
Prevalence	-0.13746621	1.0000000	0.95813468	0.96568034	0.64002950	0.11
Lower.CI	-0.07863304	0.9581347	1.00000000	0.85132455	0.67690938	0.21
Upper.CI	-0.17389486	0.9656803	0.85132455	1.00000000	0.56480277	0.02
Year	0.02851671	0.6400295	0.67690938	0.56480277	1.00000000	0.29
Numerator_ASD	0.82429404	0.1121787	0.21429644	0.02005452	0.29628163	1.00
Numerator_NonASD	0.99999025	-0.1392238	-0.08080949	-0.17516773	0.02638864	0.82
Proportion	-0.13735462	0.9999677	0.95851437	0.96524017	0.64020778	0.11
Chi_Wilson_Corrected_Lower.CI	-0.08734046	0.9761979	0.99597141	0.88837741	0.67167964	0.19
Chi_Wilson_Corrected_Upper.CI	-0.17380524	0.9798117	0.88384420	0.99561482	0.58775086	0.03

In [351]:

```
# -----
# Visualise Correlation Matrix
# -----
if(!require(corrplot)){install.packages("corrplot")}
library('corrplot')
```

Loading required package: corrplot
corrplot 0.84 loaded

In [352]:

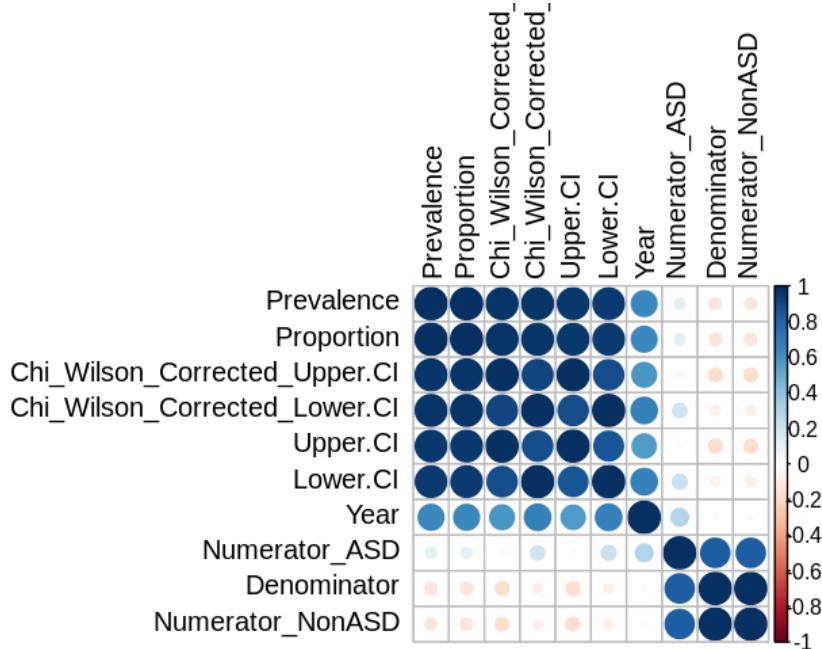
```
# Sort on decreasing correlations with Prevalence
cor_table_sorted <- as.matrix(sort(cor_table[, 'Prevalence'], decreasing = TRUE))
#
cor_table_sorted
```

Prevalence	1.0000000
Proportion	0.9999677
Chi_Wilson_Corrected_Upper.CI	0.9798117
Chi_Wilson_Corrected_Lower.CI	0.9761979
Upper.CI	0.9656803
Lower.CI	0.9581347
Year	0.6400295
Numerator_ASD	0.1121787
Denominator	-0.1374662
Numerator_NonASD	-0.1392238

```
In [353]: # Select correlations variables based on threshold:  
#cor_var_high <- names(which(apply(cor_table_sorted, 1, function(x) abs(x)>0.2  
cor_var_high <- names(which(apply(cor_table_sorted, 1, function(x) abs(x)>0.05  
#  
cor_var_high
```

```
'Prevalence' 'Proportion' 'Chi_Wilson_Corrected_Upper.CI' 'Chi_Wilson_Corrected_Lower.CI'  
'Upper.CI' 'Lower.CI' 'Year' 'Numerator_ASD' 'Denominator' 'Numerator_NonASD'
```

```
In [354]: # Visualise:  
cor_table_plot <- cor_table[cor_var_high, cor_var_high]  
# cor_table_plot  
#  
corrplot(cor_table_plot, tl.col="black", tl.pos = "lt")
```



[https://github.com/dd-consulting_\(https://github.com/dd-consulting\)](https://github.com/dd-consulting_(https://github.com/dd-consulting))

