DS3 Workshop: Exploratory Data Analysis

Shahriar Shams

Nov 15, 2023

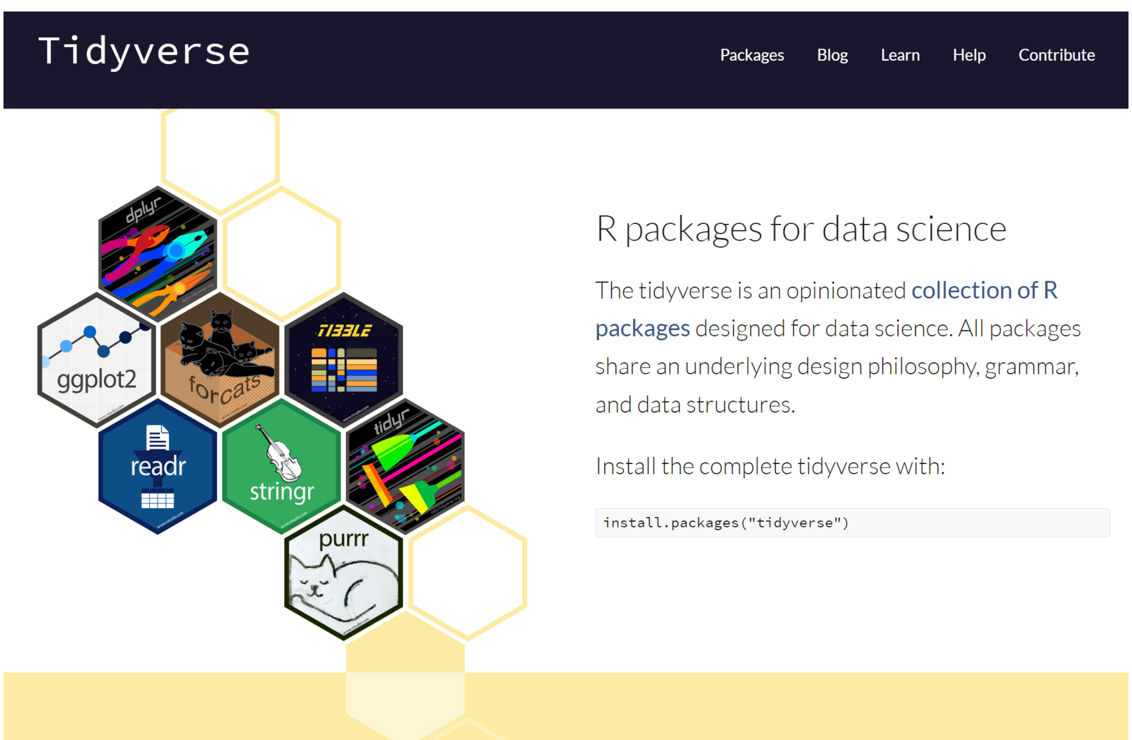
Overview

In this workshop we will cover some basic exploratory data analysis and missing value imputation techniques. Topics to be covered includes

- Getting familiar with our data
  - Creating summary tables
  - Creating box-plots, histograms
  - Creating scatter plots
  - Creating correlation plots
- Univariate analysis using simple linear regression
- Dealing with missing values

Introduction to tidyverse package

tidyverse (https://www.tidyverse.org/) is a collection of R packages that are extremely helpful for basic to advanced level data science projects. Out of the collection of packages, some of the well known packages are *dplyr, stringr, ggplot2* etc.



Getting familiar with our data

Loading data and printing some observations

```r
# reading the .csv file and naming it "d"
d = read.csv("Salary.csv", stringsAsFactors = TRUE)

# printing the first 6 observations of the data set
head(d)
```

```
##   Age Gender Education.Level         Job.Title Years.of.Experience Salary
## 1  32   Male               1 Software Engineer                   5  90000
## 2  28 Female               2      Data Analyst                   3  65000
## 3  45   Male               3           Manager                  15 150000
## 4  36 Female               1   Sales Associate                   7  60000
## 5  52   Male               2          Director                  20 200000
## 6  29   Male               1 Marketing Analyst                   2  55000
##   Country     Race Senior
## 1      UK    White      0
## 2     USA Hispanic      0
## 3  Canada    White      1
## 4     USA Hispanic      0
## 5     USA    Asian      0
## 6     USA Hispanic      0
```

```r
#
# printing the last 6 observations of the dataset
tail(d)
```

```
##      Age Gender Education.Level             Job.Title Years.of.Experience
## 6679  37   Male               1  Sales Representative                   6
## 6680  49 Female               3 Director of Marketing                  20
## 6681  32   Male               0       Sales Associate                   3
## 6682  30 Female               1     Financial Manager                   4
## 6683  46   Male               2     Marketing Manager                  14
## 6684  26 Female               0       Sales Executive                   1
##      Salary   Country       Race Senior
## 6679  75000    Canada      Asian      0
## 6680 200000        UK      Mixed      0
## 6681  50000 Australia Australian      0
## 6682  55000     China    Chinese      0
## 6683 140000     China     Korean      0
## 6684  35000    Canada      Black      0
```

- These snapshots gives us the first first impression of the data.

Creating a basic summary

```r
# printing the dimension of the table (# of rows and columns)
dim(d)
```

```
## [1] 6684    9
```

```r
# Creating an overall summary of the data
summary(d)
```

```
##       Age           Gender     Education.Level                    Job.Title   
##  Min.   :21.00   Female:3013   Min.   :0.000   Software Engineer        : 809  
##  1st Qu.:28.00   Male  :3671   1st Qu.:1.000   Data Scientist           : 515  
##  Median :32.00                 Median :1.000   Data Analyst             : 391  
##  Mean   :33.61                 Mean   :1.622   Software Engineer Manager: 376  
##  3rd Qu.:38.00                 3rd Qu.:2.000   Product Manager          : 323  
##  Max.   :62.00                 Max.   :3.000   Project Engineer         : 317  
##                                                (Other)                  :3953  
##  Years.of.Experience     Salary            Country             Race     
##  Min.   : 0.000      Min.   :   350   Australia:1335   White     :1957  
##  1st Qu.: 3.000      1st Qu.: 70000   Canada   :1322   Asian     :1599  
##  Median : 7.000      Median :115000   China    :1339   Korean    : 457  
##  Mean   : 8.078      Mean   :115307   UK       :1332   Australian: 452  
##  3rd Qu.:12.000      3rd Qu.:160000   USA      :1356   Chinese   : 443  
##  Max.   :34.000      Max.   :250000                    Black     : 435  
##                                                        (Other)   :1341  
##      Senior      
##  Min.   :0.0000  
##  1st Qu.:0.0000  
##  Median :0.0000  
##  Mean   :0.1435  
##  3rd Qu.:0.0000  
##  Max.   :1.0000  
## 
```

- For each numeric variable (including categorical variables that are coded using numeric numbers), this summary will produce the following summaries:
  - Min: The minimum value.
  - 1st Qu: The value of the first quartile (25th percentile).
  - Median: The median value.
  - Mean: The mean value.
  - 3rd Qu: The value of the third quartile (75th percentile).
  - Max: The maximum value.
- For each categorical variable it will show a portion of the categories and their corresponding frequencies.
- If there are any missing observations, this summary will also show us that. In this example, it is a complete data hence and hence there are no summary for missing-ness.

Summary using the *tidyverse* package

- Let's create the summary, but this time using the *tidyverse* package.

```
library(tidyverse)

glimpse(d)
```

```
## Rows: 6,684
## Columns: 9
## $ Age                 <dbl> 32, 28, 45, 36, 52, 29, 42, 31, 26, 38, 29, 48, 35~
## $ Gender              <fct> Male, Female, Male, Female, Male, Male, Female, Ma~
## $ Education.Level     <int> 1, 2, 3, 1, 2, 1, 2, 1, 1, 3, 2, 1, 1, 2, 1, 1, 2,~
## $ Job.Title           <fct> Software Engineer, Data Analyst, Manager, Sales As~
## $ Years.of.Experience <dbl> 5, 3, 15, 7, 20, 2, 12, 4, 1, 10, 3, 18, 6, 14, 2,~
## $ Salary              <dbl> 90000, 65000, 150000, 60000, 200000, 55000, 120000~
## $ Country             <fct> UK, USA, Canada, USA, USA, USA, USA, China, China,~
## $ Race                <fct> White, Hispanic, White, Hispanic, Asian, Hispanic,~
## $ Senior              <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
```

- In this summary, not only we can see the different variable types, but also can see some of the first few observations for each of the variables.

Summary tables (frequency table) for categorical variables

```
table(d$Gender)
```

```
##
## Female   Male
##   3013   3671
```

```
table(d$Education.Level)
```

```
##
##    0    1    2    3
##  436 3021 1858 1369
```

```
table(d$Country)
```

```
##
## Australia    Canada     China        UK       USA
##      1335      1322      1339      1332      1356
```

```
table(d$Race)
```

```
##
## African American            Asian       Australian            Black
##              352             1599              452              435
##          Chinese         Hispanic           Korean            Mixed
##              443              322              457              334
##            Welsh            White
##              333             1957
```

```r
table(d$Senior)
```

```
##
##    0    1
## 5725  959
```

- this *table()* command works for both categorical and numeric variables.

Creating ***factor*** variables

- Anytime a categorical variable is coded in numeric numbers (e.g. "Education.Level" and "Senior" in this data set), R (or any software) does not know that it is a representation of a categorical variable.
- Hence we need to convert them into factors. The variables will look exactly the same, just in the background R will know that 0 and 1 are not really 0 and 1, rather they represent two categories.

```r
# Creating a new variable called E.Level and S.yesno
d = d %>% mutate(E.Level = factor(Education.Level),
                 S.yesno = factor(Senior))

head(d)
```

```
##   Age Gender Education.Level         Job.Title Years.of.Experience Salary
## 1  32   Male               1 Software Engineer                   5  90000
## 2  28 Female               2      Data Analyst                   3  65000
## 3  45   Male               3           Manager                  15 150000
## 4  36 Female               1   Sales Associate                   7  60000
## 5  52   Male               2          Director                  20 200000
## 6  29   Male               1 Marketing Analyst                   2  55000
##   Country     Race Senior E.Level S.yesno
## 1      UK    White      0       1       0
## 2     USA Hispanic      0       2       0
## 3  Canada    White      1       3       1
## 4     USA Hispanic      0       1       0
## 5     USA    Asian      0       2       0
## 6     USA Hispanic      0       1       0
```

```r
glimpse(d)
```

```
## Rows: 6,684
## Columns: 11
## $ Age                 <dbl> 32, 28, 45, 36, 52, 29, 42, 31, 26, 38, 29, 48, 35~
## $ Gender              <fct> Male, Female, Male, Female, Male, Male, Female, Ma~
## $ Education.Level     <int> 1, 2, 3, 1, 2, 1, 2, 1, 1, 3, 2, 1, 1, 2, 1, 1, 2,~
## $ Job.Title           <fct> Software Engineer, Data Analyst, Manager, Sales As~
## $ Years.of.Experience <dbl> 5, 3, 15, 7, 20, 2, 12, 4, 1, 10, 3, 18, 6, 14, 2,~
## $ Salary              <dbl> 90000, 65000, 150000, 60000, 200000, 55000, 120000~
## $ Country             <fct> UK, USA, Canada, USA, USA, USA, USA, China, China,~
## $ Race                <fct> White, Hispanic, White, Hispanic, Asian, Hispanic,~
## $ Senior              <int> 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
## $ E.Level             <fct> 1, 2, 3, 1, 2, 1, 2, 1, 1, 3, 2, 1, 1, 2, 1, 1, 2,~
## $ S.yesno             <fct> 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,~
```

Creating cross tables

```
# table of Gender and Education level
t1 = table(d$Gender, d$E.Level)
t1
```

```
##
##             0    1    2    3
##   Female  251 1198 1068  496
##   Male    185 1823  790  873
```

```
#
# Creating table with proportions
prop.table(t1)
```

```
##
##                   0          1          2          3
##   Female 0.03755236 0.17923399 0.15978456 0.07420706
##   Male   0.02767804 0.27274087 0.11819270 0.13061041
```

```
#
# proportions calculated using row totals
prop.table(t1,margin=1)
```

```
##
##                   0          1          2          3
##   Female 0.08330568 0.39761036 0.35446399 0.16461998
##   Male   0.05039499 0.49659493 0.21520022 0.23780986
```

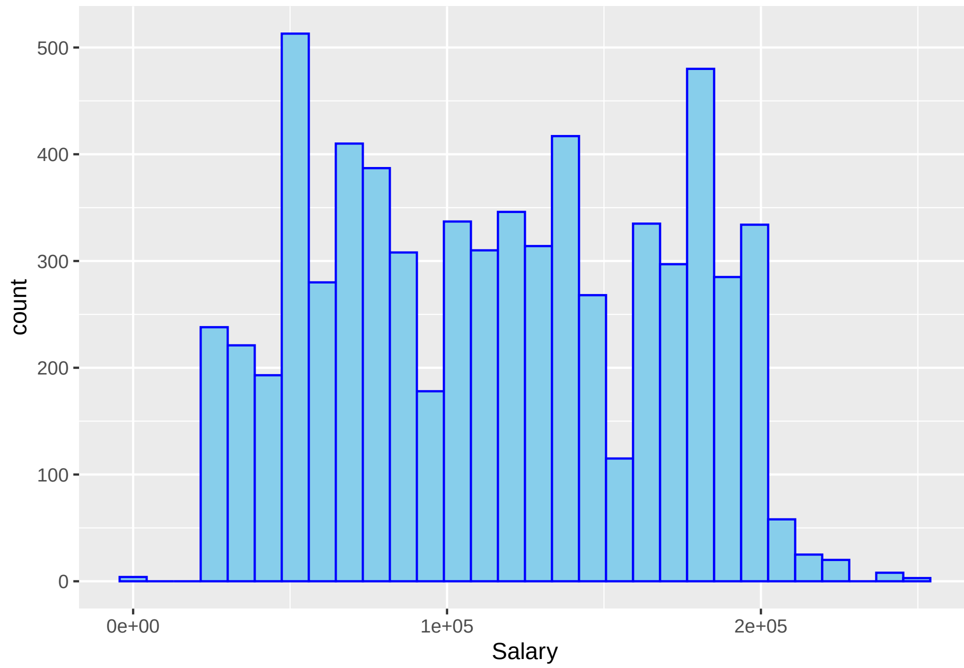
```
#
# proportions calculated using column totals
prop.table(t1,margin=2)
```

```
##
##                  0         1         2         3
##   Female 0.5756881 0.3965574 0.5748116 0.3623083
##   Male   0.4243119 0.6034426 0.4251884 0.6376917
```

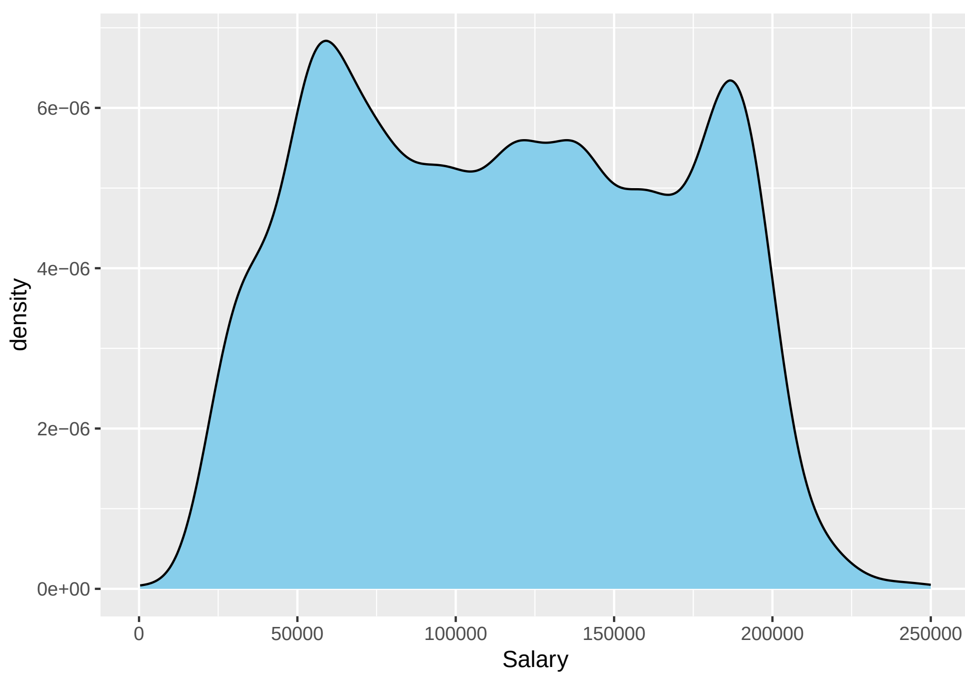
Creating histograms/density curves

- Histograms are very useful to find the overall distribution of a variable.

```r
# a histogram using the variable salary
ggplot(d, aes(x = Salary))+
       geom_histogram(colour="blue", fill="skyblue")
```




```r
#
# a density curve based on salary
ggplot(d, aes(x = Salary))+
       geom_density(fill="skyblue")
```



Creating histograms as a function of another variable

```
# Salary as a function of senior(yes vs no)
ggplot(d, aes(x = Salary, colour = S.yesno, fill= S.yesno))+
      geom_density(alpha = 0.5) # alpha value makes the graph transparent
```

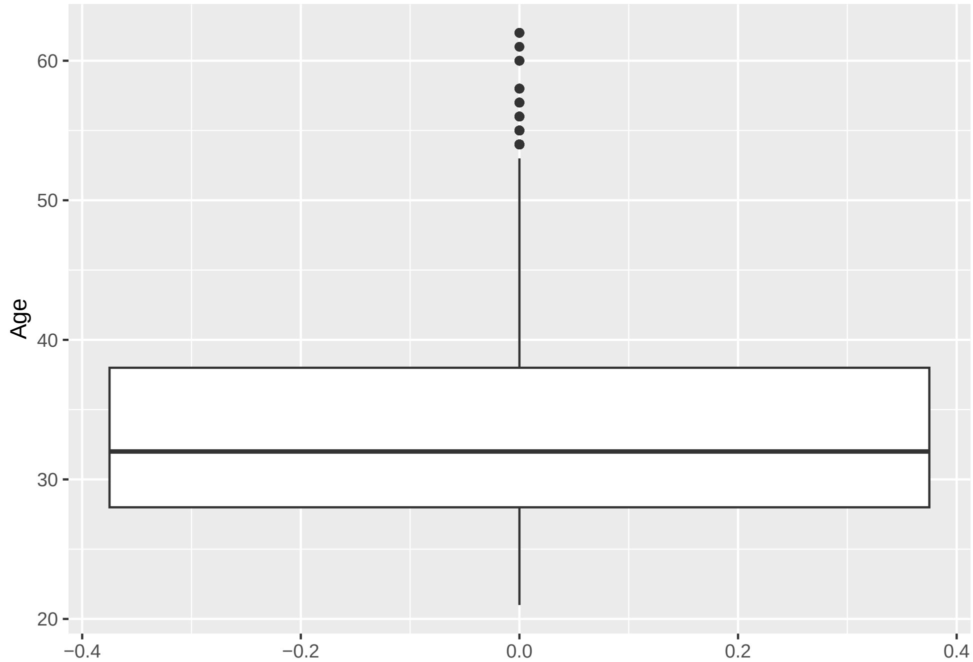

```
#
# Salary as a function of Education level
ggplot(d, aes(x = Salary, colour = E.Level, fill= E.Level))+
      geom_density(alpha = 0.5)
```

- looks like the salary variable is related to Education level and seniority.

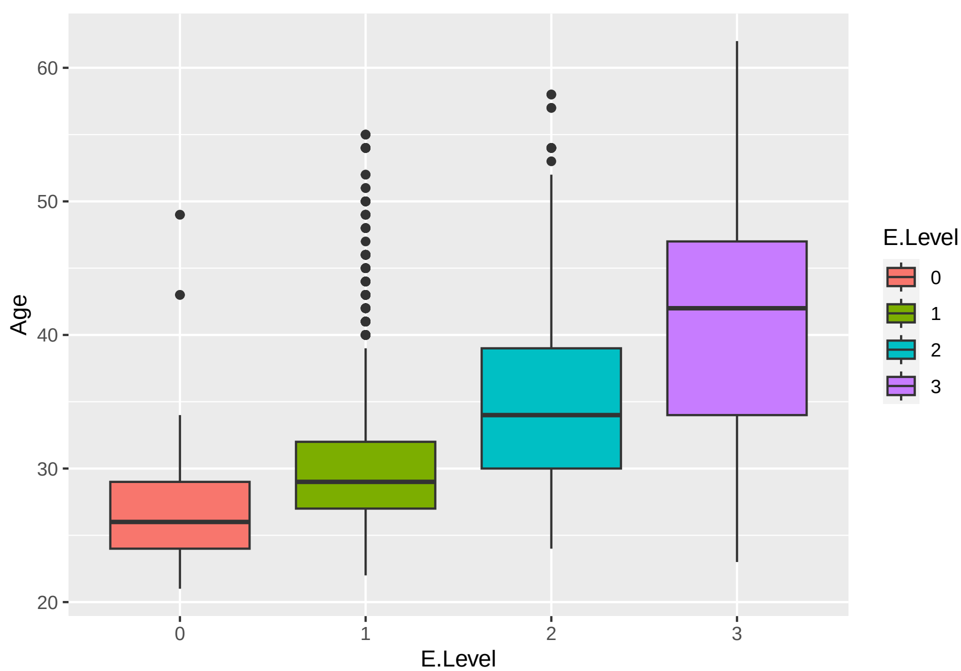
Creating boxplots

- Box plots are gives us two information: 1) distribution of the data and 2) presence of outliers

```
ggplot(d, aes(y=Age)) +
  geom_boxplot()
```




```
ggplot(d, aes(y=Age, x=E.Level, fill= E.Level)) +
  geom_boxplot()
```



Filtering data

- In the previous page we saw some outliers for the Age variable. Let's filter these observations.

```
d2 = d %>% filter(Age >55) %>% arrange(desc(Age))

dim(d2)
```

```
## [1] 39 11
```

```
head(d2)
```

```
##   Age Gender Education.Level                 Job.Title Years.of.Experience
## 1  62   Male               3 Software Engineer Manager                  19
## 2  62   Male               3 Software Engineer Manager                  20
## 3  62   Male               3 Software Engineer Manager                  19
## 4  62   Male               3 Software Engineer Manager                  20
## 5  62   Male               3 Software Engineer Manager                  19
## 6  61   Male               3 Software Engineer Manager                  20
##   Salary   Country   Race Senior E.Level S.yesno
## 1  2e+05        UK  White      0       3       0
## 2  2e+05     China Korean      0       3       0
## 3  2e+05        UK  Mixed      0       3       0
## 4  2e+05    Canada  Asian      0       3       0
## 5  2e+05 Australia  Asian      0       3       0
## 6  2e+05        UK  Welsh      0       3       0
```

Creating summary tables

- Let's create a random summary based on this filtered data

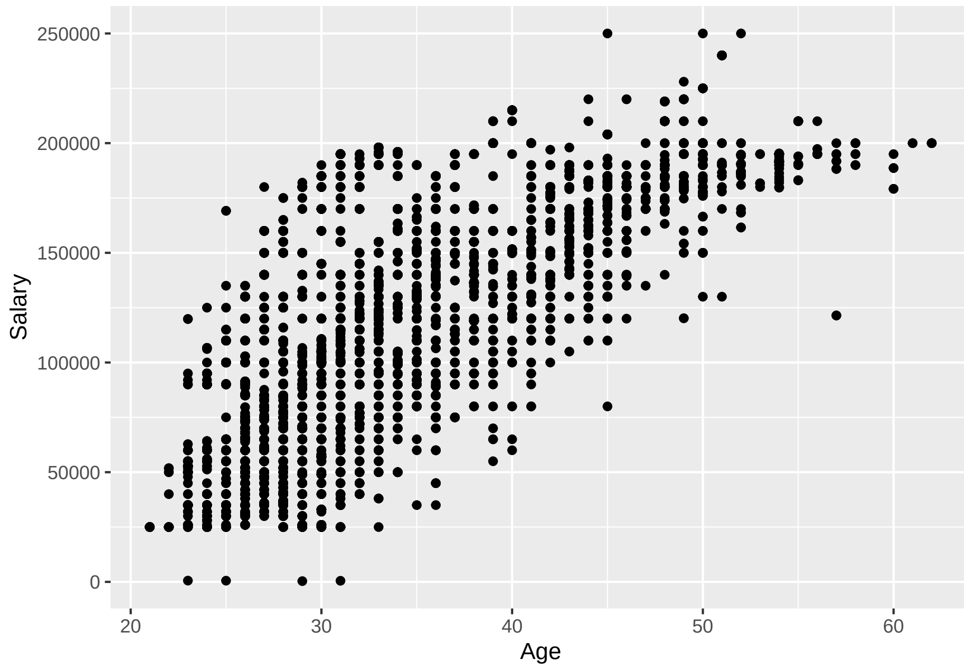
```
d2 %>%  group_by(Gender, E.Level) %>%
  summarize( Counts = n(),
             Avg_Salary = mean(Salary),
             Avg_Experience = mean(Years.of.Experience))
```

```
## # A tibble: 4 x 5
## # Groups:   Gender [2]
##   Gender E.Level Counts Avg_Salary Avg_Experience
##   <fct>  <fct>    <int>      <dbl>          <dbl>
## 1 Female 2            2    188232            33
## 2 Female 3           11    178591.           29.6
## 3 Male   2            2    190004            27
## 4 Male   3           24    197292.           17.9
```

Creating scatter plots

- Scatter plot allows us to see the relationship between two numeric variables (preferably continuous variables)

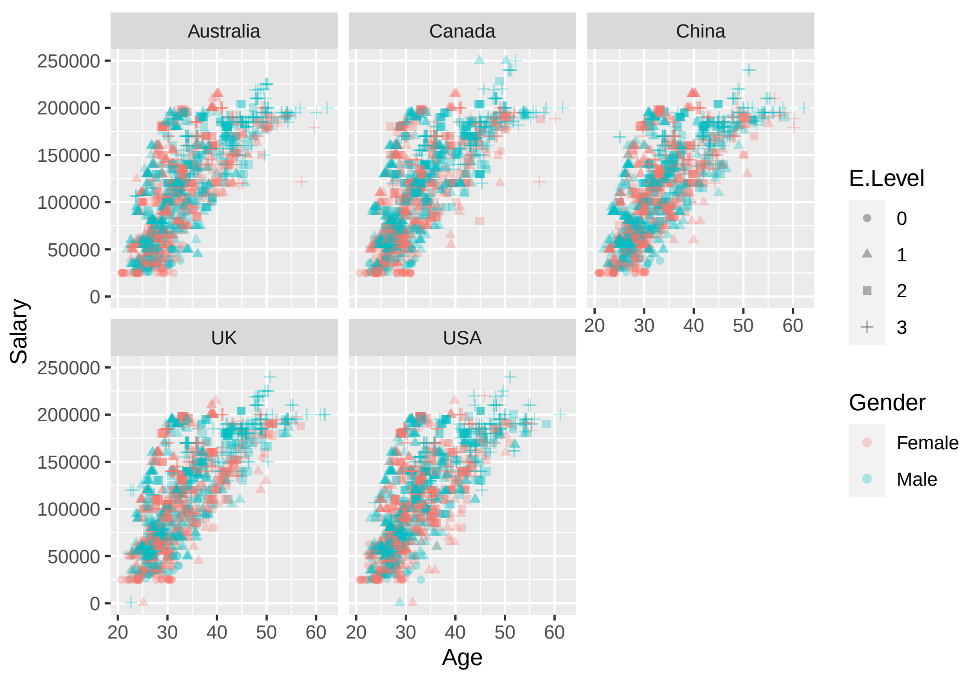
```r
# a basic scatter plot of Salary against Age
ggplot(d, aes(x=Age,y=Salary)) +
  geom_point()
```



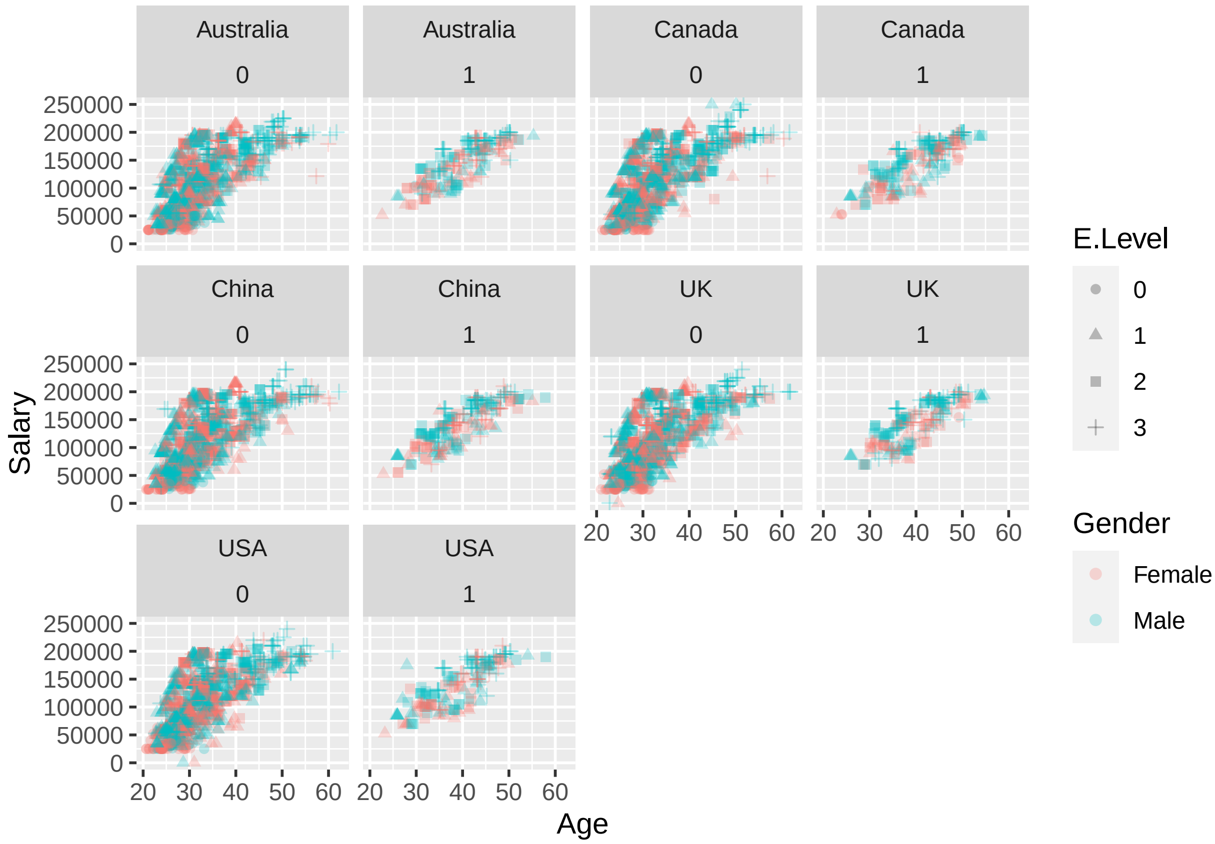

```r
#
# Salary against Age, Gender added as the color parameter
ggplot(d, aes(x=Age,y=Salary,colour=Gender)) +
  geom_jitter(alpha=0.5)
```

```r
# Salary against Age
# with Gender as the color, Education level as the shape
ggplot(d, aes(x=Age,y=Salary,colour=Gender, shape = E.Level)) +
  geom_jitter(alpha=0.3)
```


```r
# Dividing our plots by different country using facet_wrap
ggplot(d, aes(x=Age,y=Salary,colour=Gender, shape = E.Level)) +
  geom_jitter(alpha=0.3)+
  facet_wrap(~Country)
```



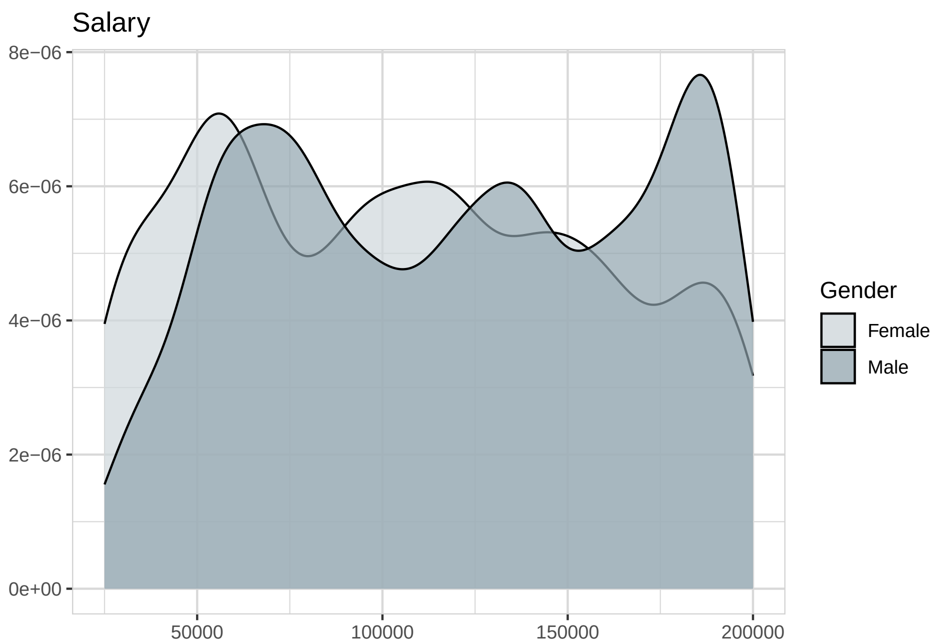
```r
# use of multiple variables in facet_wrap
ggplot(d, aes(x=Age,y=Salary,colour=Gender, shape = E.Level)) +
  geom_jitter(alpha=0.25)+
  facet_wrap(~Country+S.yesno)
```



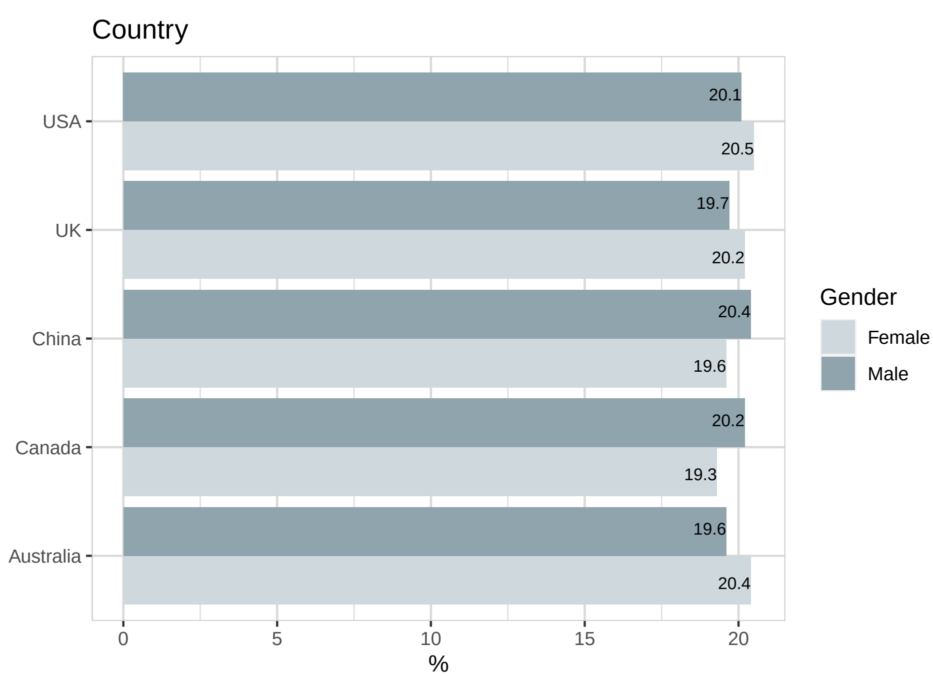
Exploratory data anlysis using *explore* package

- The best way to use this package is in an interactive R session.
- We can also create some default summaries

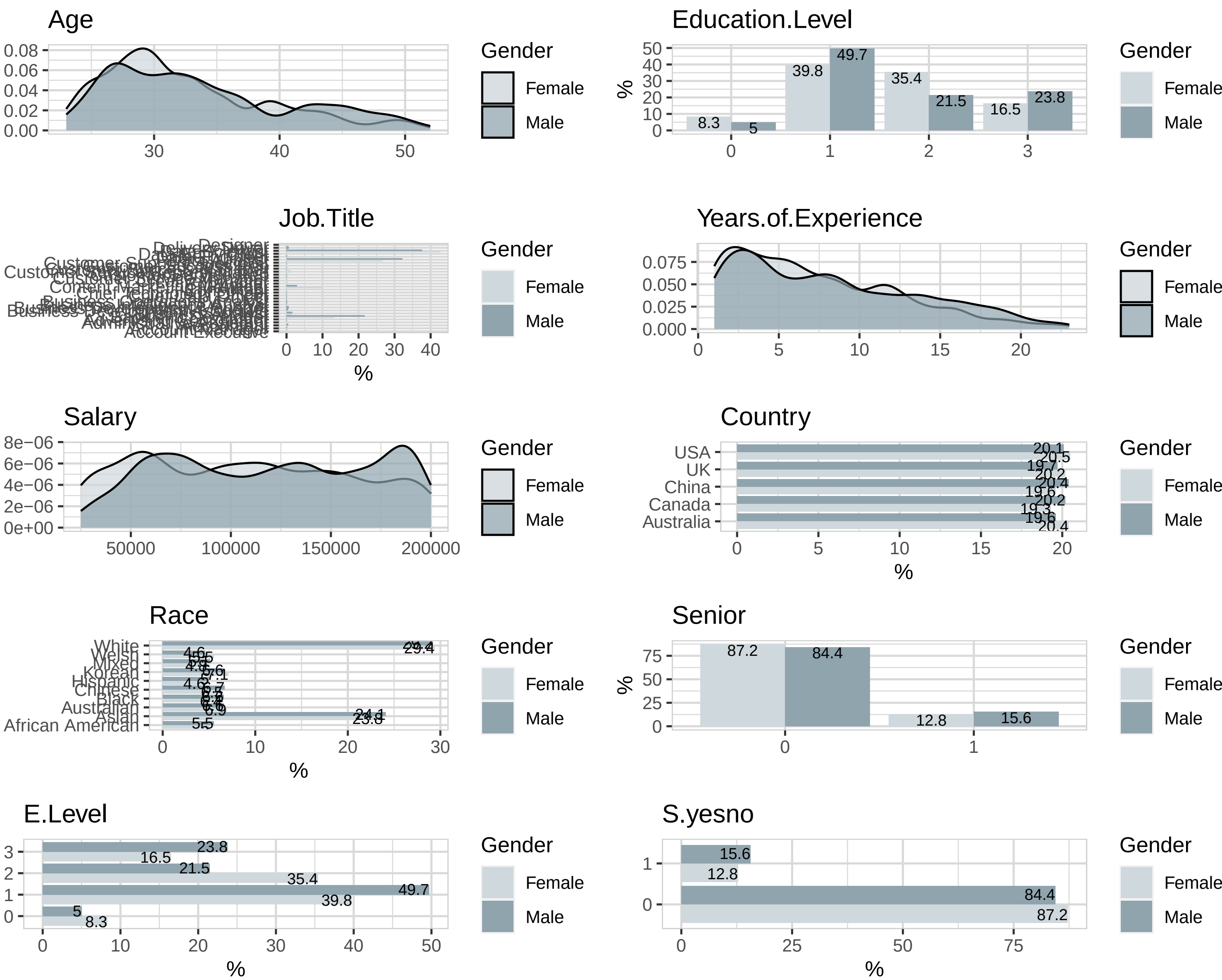
```
library(explore)

d %>%  explore(Salary,target=Gender)
```




```
d %>% explore(Country, target = Gender)
```



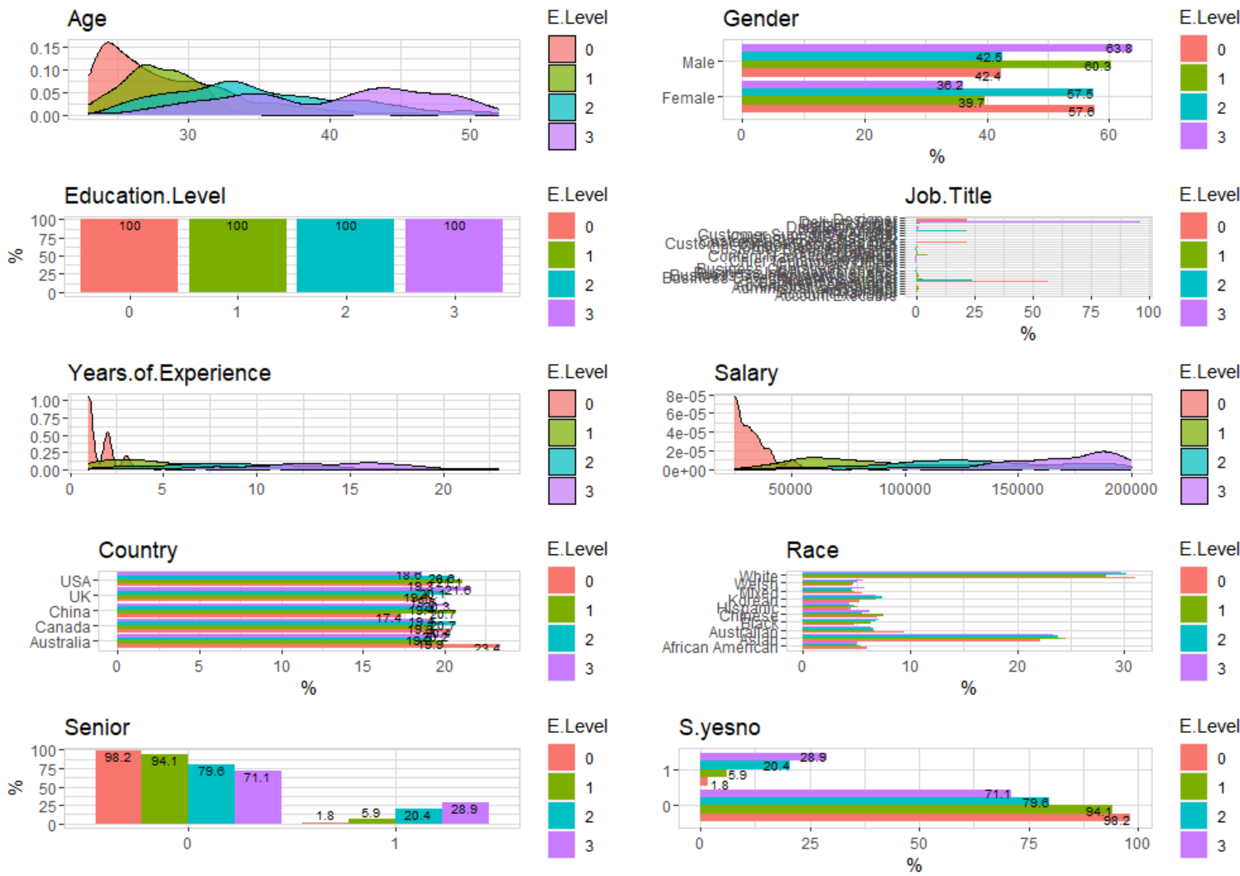
```r
d %>%  explore_all(target= Gender)
```



- In interactive sessions these plots show up in colors here is an example

```
d %>%  explore_all(target= E.Level)
```

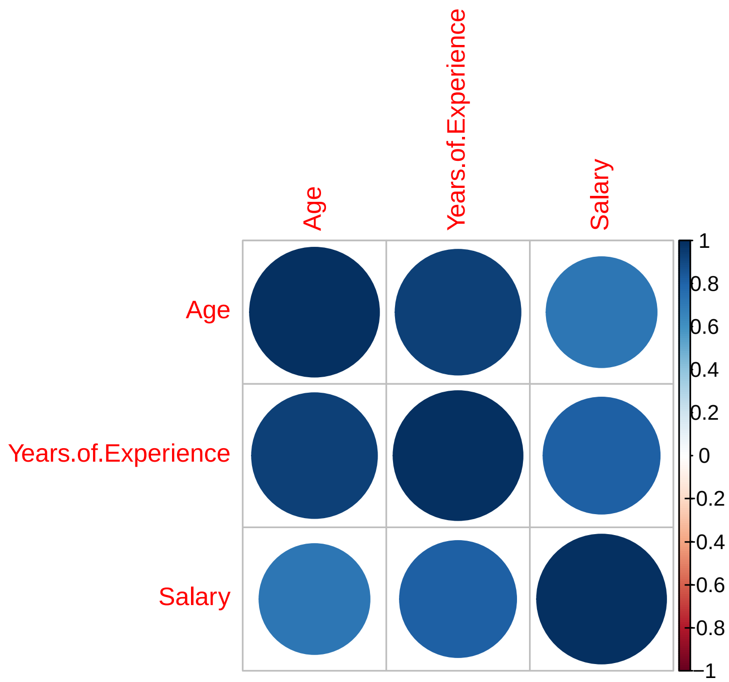
the above line produces this following plot in an interactive session.



Creating correaltion plots

- Correlation looks at the linear relationship between continuous numeric variables

```
d3 = d %>% select(Age, Years.of.Experience, Salary)

library(corrplot)
corrplot(cor(d3))
```




```
corrplot(cor(d3),method="number")
```

Exercise-1

- Load the Boston dataset by running these following two lines

```
library(MASS)
d.boston = Boston
```

1. Crete a quick summary of the dataset that includes how many observations are there, how many variables are there, what are the variable types etc.
2. Create a density curve for the "medv" variable by making separate densities for the "chas" variable values.
3. Create a scatter plot of "medv" against "lstat" while using "chas" as the *colour* and the *shape* parameter.
4. Create a correlation plot using all the variables of the data.

Univariate analysis using simple linear regression

- After we are done with our exploration using graphs and tables, we can use a bit more advanced tool (which is a bit more objective) to check the relations.
- Simple Linear Regression is one such tool.

```r
m = lm(Salary~Age, data=d)
summary(m)
```

```
##
## Call:
## lm(formula = Salary ~ Age, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -112287  -26899   -6645   22150   98165
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -54876.15    2008.02  -27.33   <2e-16 ***
## Age           5063.39      58.27   86.89   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36190 on 6682 degrees of freedom
## Multiple R-squared:  0.5305, Adjusted R-squared:  0.5304
## F-statistic:  7550 on 1 and 6682 DF,  p-value: < 2.2e-16
```

```r
#
#
m = lm(Salary~Gender, data = d)
summary(m)
```

```
##
## Call:
## lm(formula = Salary ~ Gender, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -121046  -47789   -1396   47111  128604
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 107889.0      954.3  113.06   <2e-16 ***
## GenderMale   13506.7     1287.7   10.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52380 on 6682 degrees of freedom
## Multiple R-squared:  0.0162, Adjusted R-squared:  0.01605
## F-statistic:   110 on 1 and 6682 DF,  p-value: < 2.2e-16
```

```r
m = lm(Salary~E.Level, data=d)
summary(m)
```

```
##
## Call:
## lm(formula = Salary ~ E.Level, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -165072  -30078   -5078   24917  154917
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    34416       1914   17.98   <2e-16 ***
## E.Level1       60667       2048   29.63   <2e-16 ***
## E.Level2       95663       2127   44.98   <2e-16 ***
## E.Level3      131236       2198   59.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39970 on 6680 degrees of freedom
## Multiple R-squared:  0.4273, Adjusted R-squared:  0.4271
## F-statistic:  1662 on 3 and 6680 DF,  p-value: < 2.2e-16
```

```r
#
m = lm(Salary~Country, data = d)
summary(m)
```

```
##
## Call:
## lm(formula = Salary ~ Country, data = d)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115370  -46326      75   44789  133545
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   114925.5     1445.3  79.518   <2e-16 ***
## CountryCanada   1529.6     2049.0   0.747    0.455
## CountryChina    1357.1     2042.4   0.664    0.506
## CountryUK        994.5     2045.1   0.486    0.627
## CountryUSA     -1926.7     2036.0  -0.946    0.344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52810 on 6679 degrees of freedom
## Multiple R-squared:  0.0005868,  Adjusted R-squared:  -1.169e-05
## F-statistic: 0.9805 on 4 and 6679 DF,  p-value: 0.4168
```

```r
table(d$Country)
```

```
##
```

```
## Australia     Canada      China         UK        USA
##       1335       1322       1339       1332       1356
```

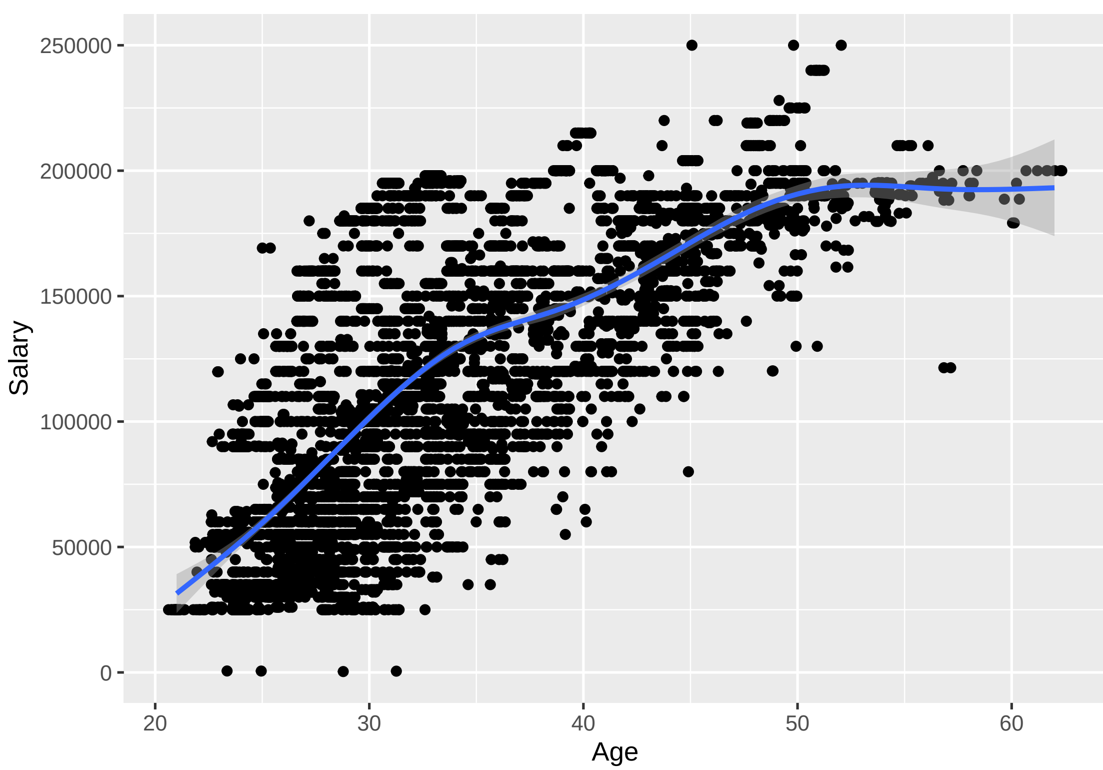
comments based on the outputs of these four models:

- Looks like Age, Gender, E.Level are all significantly associated with Salary.
- But there is not a lot of variation in Salaries among the different countries.
- Of course you can continue to fit these models for all other explanatory variables.

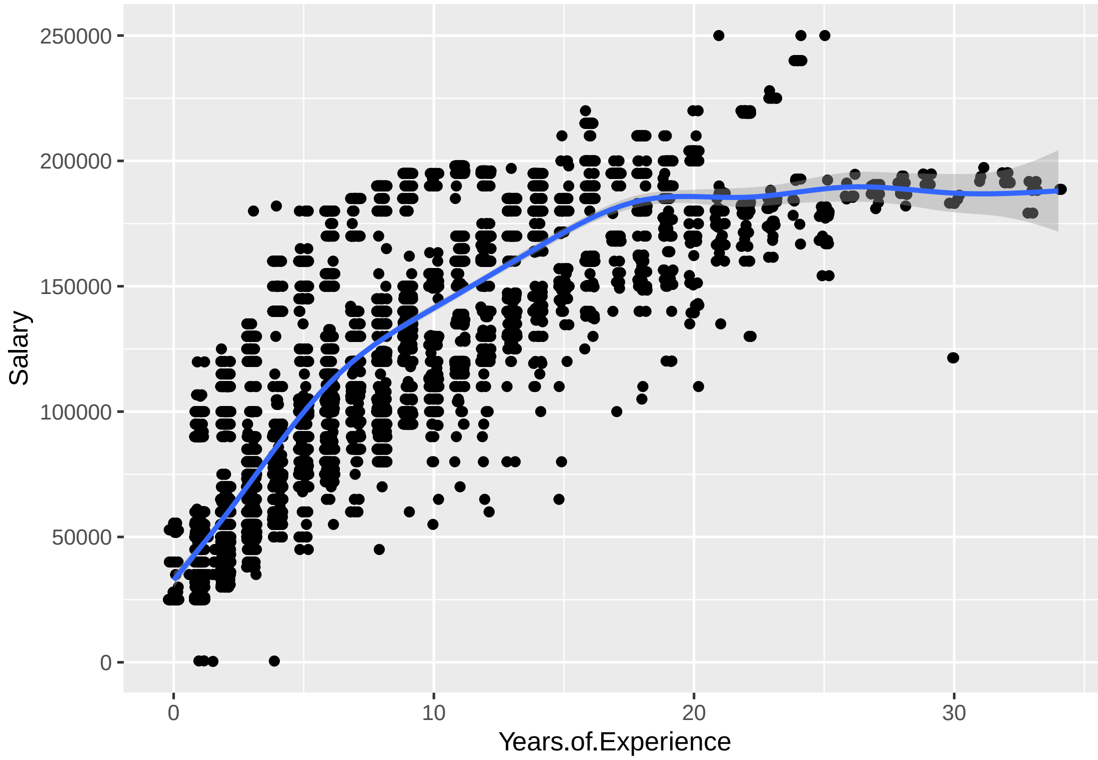
Using ggplot2 to graph the relationship

- geom_smooth() adds smooth non-linear curves on top of any scatter plot.
- If we want to restrict it to certain model (e.g. linear), we can put that formula inside the geom_smooth() command.

```
ggplot(d, aes(x=Age, y = Salary))+
  geom_jitter()+
  geom_smooth()
```



```r
ggplot(d, aes(x=Years.of.Experience, y = Salary))+
  geom_jitter()+
  geom_smooth()
```



- These pictures indicate that when fitting our final model, we should consider nor linear functions of Age or Year of Experience.

Dealing with missing values

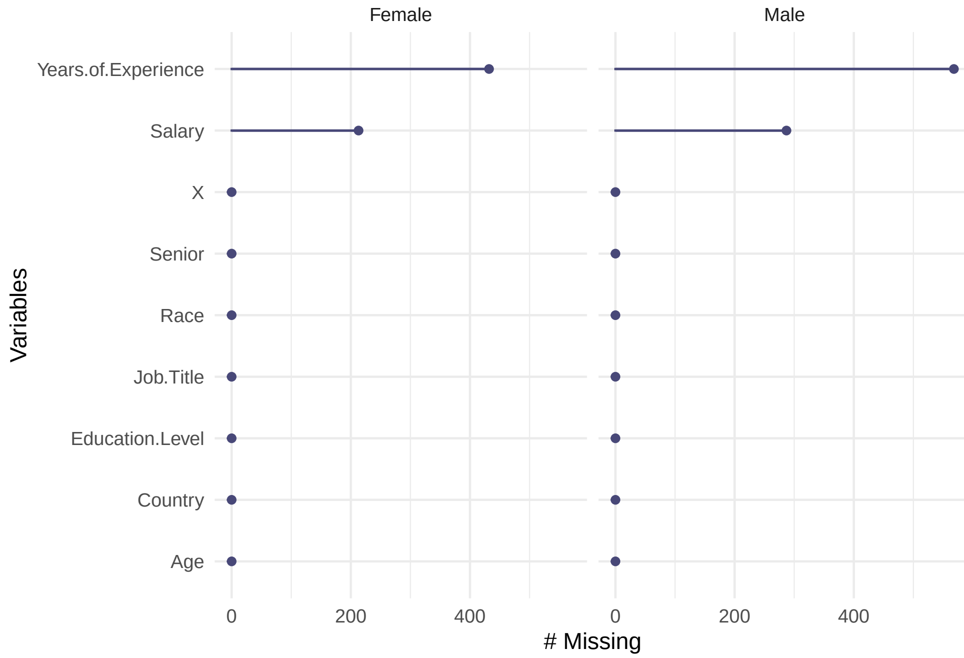
- Missing value is a common feature in almost all real world data.
- For numerous reasons, values in one of the variable/feature or more than one variable/feature can be missing.
- A missing observation can be a result of the respondent not answering the question, or for any systematic reason, or simply a data entry error.
- When missing values are present, we can either remove them or impute them.

Summary of missing-ness using the ***naniar*** package

```r
# reading a different version of the dataset with some missing values
d.miss = read.csv(file="salary_with_missing_values.csv")



library(naniar)

miss_var_summary(d.miss)
```

```
## # A tibble: 10 x 3
##    variable            n_miss pct_miss
##    <chr>                <int>    <dbl>
##  1 Years.of.Experience   1000    15.0
##  2 Salary                 500     7.48
##  3 X                        0     0
##  4 Age                      0     0
##  5 Gender                   0     0
##  6 Education.Level          0     0
##  7 Job.Title                0     0
##  8 Country                  0     0
##  9 Race                     0     0
## 10 Senior                   0     0
```

```r
gg_miss_var(d.miss, facet = Gender)
```



Option-1: removing rows

- This is a bad option.

```
dim(d.miss)
```

```
## [1] 6684   10
```

```
#
# omitting the rows which has any missing value
d2 = na.omit(d.miss)
dim(d2)
```

```
## [1] 5258   10
```

Option-2: removing columns

```
d2 = d.miss %>% select(-Years.of.Experience)

names(d2)
```

```
## [1] "X"               "Age"             "Gender"          "Education.Level"
## [5] "Job.Title"       "Salary"          "Country"         "Race"
## [9] "Senior"
```

- This is also a bad option and not recommended!

Option-3: Imputing with the average

```
mean(d.miss$Years.of.Experience, na.rm=T) # overall average
```

```
## [1] 8.035626
```

```r
d2 = d.miss %>%
  mutate(NewYOE = case_when(
    is.na(Years.of.Experience) ~ mean(Years.of.Experience,na.rm=T),
    !is.na(Years.of.Experience) ~ Years.of.Experience) )

library(knitr)
kable(d2[1:6,-c(1:5)])
```

Years.of.Experience	Salary	Country	Race	Senior	NewYOE
5	90000	UK	White	0	5.000000
3	65000	USA	Hispanic	0	3.000000
NA	150000	Canada	White	1	8.035626
7	60000	USA	Hispanic	0	7.000000
20	200000	USA	Asian	0	20.000000
2	55000	USA	Hispanic	0	2.000000

Option-4: Imputing using a regression model

- This is probably the **best** option compared to the previous ones.

```r
library(simputation)

d.miss = d.miss %>% mutate(YOE = Years.of.Experience)

d2 <- impute_lm(d.miss,
                YOE ~ Age + Salary + Gender + Education.Level+
                  Country + Race + Senior)

kable(d2[1:5,-c(1:5)])
```

Years.of.Experience	Salary	Country	Race	Senior	YOE
5	90000	UK	White	0	5.00000
3	65000	USA	Hispanic	0	3.00000
NA	150000	Canada	White	1	15.72772
7	60000	USA	Hispanic	0	7.00000
20	200000	USA	Asian	0	20.00000

Exercise-2

1. load the "salary_with_missing_values.csv" to R/Rstudio.
2. Install and load the *simputation* package
3. Create an imputed dataset where the missing "Salary" values are imputed as a function of the rest of the variables.

References

Here are few good resources:

- https://r4ds.had.co.nz/
- https://www.rstudio.com/wp-content/uploads/2015/02/data-wrangling-cheatsheet.pdf
- https://ggplot2.tidyverse.org/index.html
- https://cran.r-project.org/web/packages/explore/vignettes/explore_titanic.html
- https://cran.r-project.org/web/packages/simputation/vignettes/intro.html
- https://r-project.ro/conference2018/presentations/simputation_presentation.pdf

{Good luck with your future Exploratory Data Analysis!}