THE UNIVERSITY OF TEXAS AT AUSTIN
McCOMBS SCHOOL OF BUSINESS

Diagnostics & Transformations 2

Lecture 13

STA 371G

Newly hired manager salaries

Newly hired manager salaries

- Salary (response)
- Manager Rating
- Years of Experience
- NUmber of years since graduation
- Origin (internal or external hire)

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Most common issues are:

- Outliers (e.g. incorrect entries, anomalies)

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Most common issues are:

- Outliers (e.g. incorrect entries, anomalies)
- Missing data

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Most common issues are:

- Outliers (e.g. incorrect entries, anomalies)
- Missing data
- Multicollinearity

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Most common issues are:

- Outliers (e.g. incorrect entries, anomalies)
- Missing data
- Multicollinearity
- Violation of LINE assumptions

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Most common issues are:

- Outliers (e.g. incorrect entries, anomalies)
- Missing data
- Multicollinearity
- Violation of LINE assumptions

Data issues

It is extremely rare that the data could be used right away without any clean up.

Data scients report that they spend 70% of their time on getting and cleaning the data. Only 30% is for statistical analysis.

Most common issues are:

- Outliers (e.g. incorrect entries, anomalies)
- Missing data
- Multicollinearity
- Violation of LINE assumptions

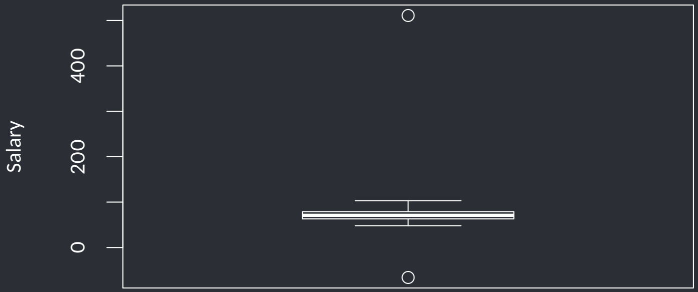
It is almost always necessary to explore and clean the data before doing the modeling

Boxplots are commonly used to detect the outliers.

Boxplots are commonly used to detect the outliers.
Let’s start with looking into the salary column.

Boxplots are commonly used to detect the outliers.
Let's start with looking into the salary column.

```
boxplot(manager$Salary, ylab='Salary')
```



There is a negative entry, and a very large one. We need to investigate these.

Exploring the data: Outliers

```
manager[manager$Salary>200,]

# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1    511        6.1        2            2 Internal

manager[manager$Salary<0,]

# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1    -66        5.7        1            2 Internal
```

Exploring the data: Outliers

```
manager[manager$Salary>200,]
```

```
# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1    511        6.1        2            2 Internal
```

```
manager[manager$Salary<0,]
```

```
# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1    -66        5.7        1            2 Internal
```

These are probably just incorrect entries.

Exploring the data: Outliers

```
    manager[manager$Salary>200,]

# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1    511        6.1        2            2 Internal

    manager[manager$Salary<0,]

# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1    -66        5.7        1            2 Internal
```

These are probably just incorrect entries.
Try to correct the data whenever you can. If not possible, we will omit them.

Exploring the data: Outliers

```
mclean <- manager[manager$Salary>0 & manager$Salary<200,]
```

Exploring the data: Outliers

```r
mclean <- manager[manager$Salary>0 & manager$Salary<200,]
```

Select the subset of the data where the salary is between 0 and 200K.

Exploring the data: Outliers

```
boxplot(mclean$YearsExp, ylab='Years of Experience')
```

Exploring the data: Outliers

Assume we somehow know that 99 is a code for missing entry in the Years of Experience data.

Exploring the data: Outliers

Assume we somehow know that 99 is a code for missing entry in the Years of Experience data.
Let’s propoerly label all 99s as “NA” (Not Available).

Exploring the data: Outliers

Assume we somehow know that 99 is a code for missing entry in the Years of Experience data.
Let's propoerly label all 99s as "NA" (Not Available).

```
mclean$YearsExp[mclean$YearsExp==99] <- NA
```

Exploring the data: Outliers

Assume we somehow know that 99 is a code for missing entry in the Years of Experience data.
Let's propoerly label all 99s as "NA" (Not Available).

```
mclean$YearsExp[mclean$YearsExp==99] <- NA
```

Now, what? We have missing entries our data!

Exploring the data: Outliers

Assume we somehow know that 99 is a code for missing entry in the Years of Experience data.
Let's propoerly label all 99s as "NA" (Not Available).

```
mclean$YearsExp[mclean$YearsExp==99] <- NA
```

Now, what? We have missing entries our data!
Great.

Exploring the data: Missing entries

Let’s see if we have other missing data.

Exploring the data: Missing entries

Let's see if we have other missing data.

```
mclean[!complete.cases(mclean),]
```

```
# A tibble: 4 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1     75         NA        8            8 Internal
2     81         NA        9            9 External
3     73        5.9       NA            7 External
4     49        8.0        1            1     <NA>
```

Exploring the data: Missing entries

Let's see if we have other missing data.

```
mclean[!complete.cases(mclean),]
```

```
# A tibble: 4 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <int>        <int>    <chr>
1     75         NA        8            8 Internal
2     81         NA        9            9 External
3     73        5.9       NA            7 External
4     49        8.0        1            1     <NA>
```

This should not come a surprise, because it is very common to have missing entries in your data.
*If you are not seeing the last entry, it is because na.strings is set to NA (it should be empty in RStudio, and na.strings=" should be used while in read.csv command).

Exploring the data: Missing entries

There are two ways of dealing with missing data.

Exploring the data: Missing entries

There are two ways of dealing with missing data.

- Omit the rows that have missing entries in it.
- Try to predict values to fill the missing entries.

Exploring the data: Missing entries

There are two ways of dealing with missing data.

- Omit the rows that have missing entries in it.
- Try to predict values to fill the missing entries.

Omiting data is the easiest, but often not the best way; because you lose all the other information available in the same row.

Exploring the data: Missing entries

There are two ways of dealing with missing data.

- Omit the rows that have missing entries in it.
- Try to predict values to fill the missing entries.

Omiting data is the easiest, but often not the best way; because you lose all the other information available in the same row.

Let's try to fill in some estimates.

Exploring the data: Missing entries

What should we replace the “NA”s in the Manager Rating and Years of Experience columns with?

Exploring the data: Missing entries

What should we replace the “NA”s in the Manager Rating and Years of Experience columns with?

The simplest way would be to use the averages in the respective columns.

```
average_MngrRating <- mean(mclean$MngrRating, na.rm=TRUE)
mclean$MngrRating[is.na(mclean$MngrRating)] <- average_MngrRating

average_YearsExp <- mean(mclean$YearsExp, na.rm=TRUE)
mclean$YearsExp[is.na(mclean$YearsExp)] <- average_YearsExp
```

Exploring the data: Missing entries

What should we replace the “NA”s in the Manager Rating and Years of Experience columns with?

The simplest way would be to use the averages in the respective columns.

```
average_MngrRating <- mean(mclean$MngrRating, na.rm=TRUE)
mclean$MngrRating[is.na(mclean$MngrRating)] <- average_MngrRating

average_YearsExp <- mean(mclean$YearsExp, na.rm=TRUE)
mclean$YearsExp[is.na(mclean$YearsExp)] <- average_YearsExp
```

A smarter and more advanced way is to predict, e.g., what the Manager Rating would be for a person with $75K salary, 8 years of experience and who is an internal hire.

Exploring the data: Missing entries

What about the missing categorical variables?

Exploring the data: Missing entries

What about the missing categorical variables?
Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

Exploring the data: Missing entries

What about the missing categorical variables?
Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

This removes all the rows that contain missing entries (only the Origin column has missing entries in this case.)

Exploring the data: Missing entries

What about the missing categorical variables?
Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

This removes all the rows that contain missing entries (only the Origin column has missing entries in this case.)

There are also ways of predicting the missing entries in a categorical variable.

Exploring the data: Missing entries

What about the missing categorical variables?
Let's choose the easy way and omit them.

```
mclean <- na.omit(mclean)
```

This removes all the rows that contain missing entries (only the Origin column has missing entries in this case.)

There are also ways of predicting the missing entries in a categorical variable.

Or we could have treated the missing entries as a seperate level (e.g. "Unknown").

Exploring the data: Missing entries

Important things to consider:

- While dealing with the missing data, the assumption is that the data is "Missing Completely at Random" (MCAR).

Exploring the data: Missing entries

Important things to consider:

- While dealing with the missing data, the assumption is that the data is “Missing Completely at Random” (MCAR).
- If this assumption does not hold (e.g. if the missing data mostly belongs to external hires), the model will be biased.

Exploring the data: Missing entries

Important things to consider:

- While dealing with the missing data, the assumption is that the data is "Missing Completely at Random" (MCAR).
- If this assumption does not hold (e.g. if the missing data mostly belongs to external hires), the model will be biased.
- Making predictions for missing data based on available data enforces the already existing relationships between variables, therefore impacts the standard error.

Exploring the data: Missing entries

Important things to consider:

- While dealing with the missing data, the assumption is that the data is "Missing Completely at Random" (MCAR).
- If this assumption does not hold (e.g. if the missing data mostly belongs to external hires), the model will be biased.
- Making predictions for missing data based on available data enforces the already existing relationships between variables, therefore impacts the standard error.
- If a lot of data is missing (e.g. more than 5%) for a particular variable, you may have to discard the whole column.

Exploring the data: Multicollinearity

In multiple regression models, multicollinearity exists whenever two or more of the predictors in a regression model are moderately or highly correlated.

Exploring the data: Multicollinearity

In multiple regression models, multicollinearity exists whenever two or more of the predictors in a regression model are moderately or highly correlated.

It is a problem, because:

- Any conclusions based on the p-values, coeffients and confidence intervals of the highly correlated variables will be unreliable.

Exploring the data: Multicollinearity

In multiple regression models, multicollinearity exists whenever two or more of the predictors in a regression model are moderately or highly correlated.

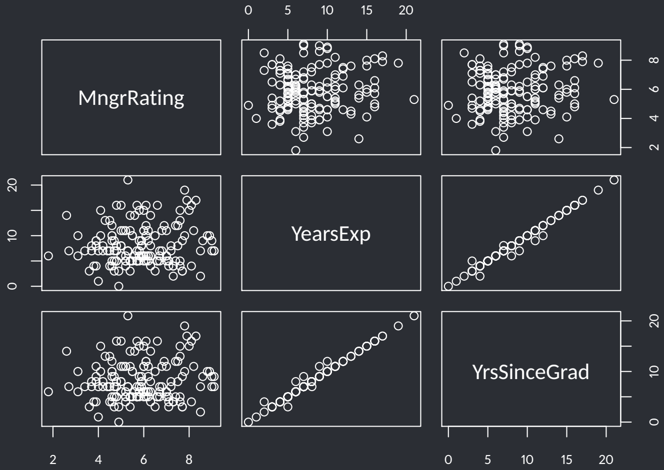
It is a problem, because:

- Any conclusions based on the p-values, coeffients and confidence intervals of the highly correlated variables will be unreliable.
- These statistics will not be stable: adding new data or predictors to the model could drastically change them.

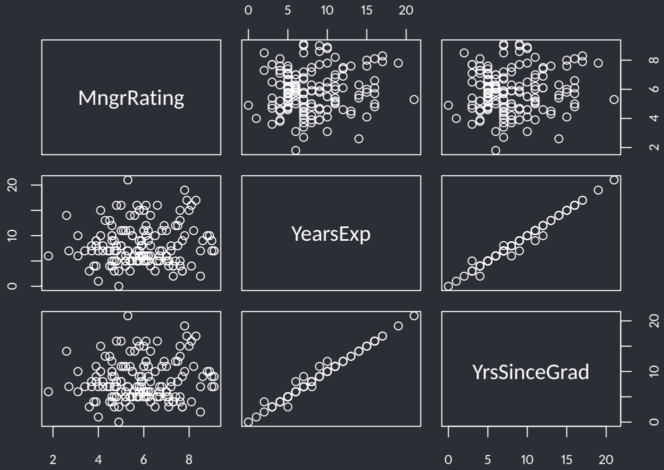
```r
model<- lm(Salary ~ MngrRating + YearsExp
                  + YrsSinceGrad + Origin, data=mclean)
summary(model)
```

```

Call:
lm(formula = Salary ~ MngrRating + YearsExp + YrsSinceGrad + 
    Origin, data = mclean)

Residuals:
     Min       1Q   Median       3Q      Max 
-19.7766  -4.2842  -0.2906   3.3266  28.2773 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     54.1521     2.6071  20.771  < 2e-16 ***
MngrRating       4.5147     0.3997  11.296  < 2e-16 ***
YearsExp        -1.5262     1.3790  -1.107 0.270203    
YrsSinceGrad     0.7692     1.3833   0.556 0.578976    
OriginInternal  -4.7314     1.3878  -3.409 0.000838 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.838 on 149 degrees of freedom
Multiple R-squared:  0.6065,Adjusted R-squared:  0.596 
F-statistic: 57.42 on 4 and 149 DF,  p-value: < 2.2e-16
```

```r
pairs(~ MngrRating+YearsExp+YrsSinceGrad, data=mclean)
```



```r
pairs(~ MngrRating+YearsExp+YrsSinceGrad, data=mclean)
```



Exploring the data: Multicollinearity

If you check the correlation between the two:

```
cor(mclean$YearsExp,mclean$YrsSinceGrad)

[1] 0.9947616
```

Exploring the data: Multicollinearity

If you check the correlation between the two:

```
cor(mclean$YearsExp,mclean$YrsSinceGrad)

[1] 0.9947616
```

Any correlation beyond 0.95 is definitely a problem.

Exploring the data: Multicollinearity

If you check the correlation between the two:

```
cor(mclean$YearsExp,mclean$YrsSinceGrad)

[1] 0.9947616
```

Any correlation beyond 0.95 is definitely a problem.

A better way to check multicollinearity:

```
library(car)
vif(model)

MngrRating     YearsExp YrsSinceGrad       Origin
  1.136002    95.954255    97.011260     1.540448
```

Drop one of the predictors that has a VIF higher than 5.

Exploring the data: Multicollinearity

If you check the correlation between the two:

```
    cor(mclean$YearsExp,mclean$YrsSinceGrad)

[1] 0.9947616
```

Any correlation beyond 0.95 is definitely a problem.

A better way to check multicollinearity:

```
  library(car)
  vif(model)

  MngrRating     YearsExp YrsSinceGrad       Origin
    1.136002    95.954255    97.011260     1.540448
```

Drop one of the predictors that has a VIF higher than 5.

Remember: Multicollinearity could exist between more than two predictors (e.g. having seperate columns with binary values for Spring, Summer, Autumn, Winter).

```r
model2<- lm(Salary ~ MngrRating + YearsExp
                  + Origin, data=mclean)
summary(model2)
```

```
Call:
lm(formula = Salary ~ MngrRating + YearsExp + Origin, data = mclean)

Residuals:
     Min       1Q   Median       3Q      Max 
-19.8115  -4.3474  -0.3964   3.3358  28.1801 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)     54.1080     2.5999  20.812  < 2e-16 ***
MngrRating       4.5309     0.3977  11.394  < 2e-16 ***
YearsExp        -0.7651     0.1687  -4.534 1.18e-05 ***
OriginInternal  -4.6467     1.3762  -3.376 0.000935 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.823 on 150 degrees of freedom
Multiple R-squared:  0.6057,Adjusted R-squared:  0.5978 
F-statistic: 76.82 on 3 and 150 DF,  p-value: < 2.2e-16
```

Checking on LINE assumptions

We always need to check on LINE assumptions.

Checking on LINE assumptions

We always need to check on LINE assumptions.
To do it at one shot, use plot() function.

- Look for a horizontal red line in the Residuals-Fitted plot for linearity.

Checking on LINE assumptions

We always need to check on LINE assumptions.
To do it at one shot, use plot() function.

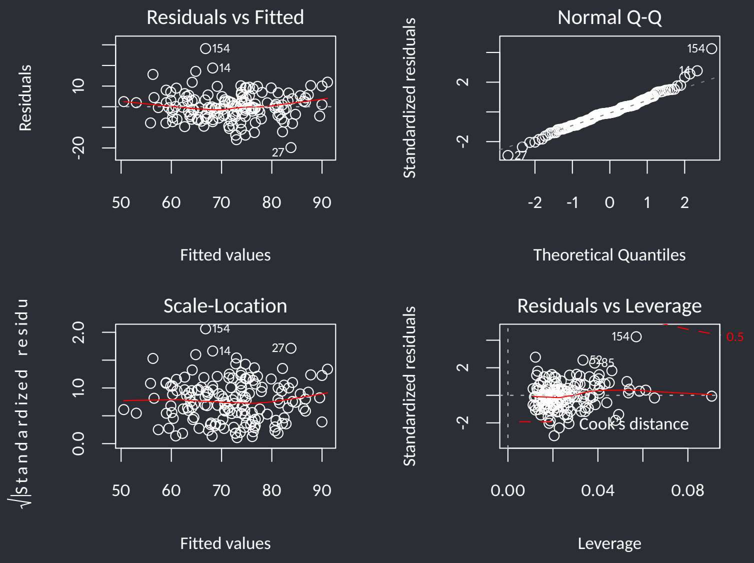
- Look for a horizontal red line in the Residuals-Fitted plot for linearity.
- Look for a horizontal red line in the Scale-Location plot for equal variance.

Checking on LINE assumptions

We always need to check on LINE assumptions.
To do it at one shot, use plot() function.

- Look for a horizontal red line in the Residuals-Fitted plot for linearity.
- Look for a horizontal red line in the Scale-Location plot for equal variance.
- Look for a straight line in the Normal Q-Q plot for normality.

```r
par(mfrow=c(2,2)) # Change the panel layout to 2 x 2
plot(model2)
```




```r
par(mfrow=c(1,1)) # Change back to 1 x 1
```

Outliers among the residuals

The data with index 154 seems to be hanging out by itself. It is worth investigating.

Outliers among the residuals

The data with index 154 seems to be hanging out by itself. It is worth investigating.

We can display the indices of all of the outliers among the residuals.

```
boxplot(resid(model2))$out
```

```
      14        27        52        85       154 
18.76901 -19.81152  17.14133  15.66079  28.18007 
```

Outliers among the residuals

The data with index 154 seems to be hanging out by itself. It is worth investigating.

We can display the indices of all of the outliers among the residuals.

```
boxplot(resid(model2))$out
```

```
      14        27        52        85       154 
18.76901 -19.81152  17.14133  15.66079  28.18007 
```

Basically, the model is not able to explain these cases very well.

Outliers among the residuals

The data with index 154 seems to be hanging out by itself. It is worth investigating.

We can display the indices of all of the outliers among the residuals.

```
boxplot(resid(model2))$out
```

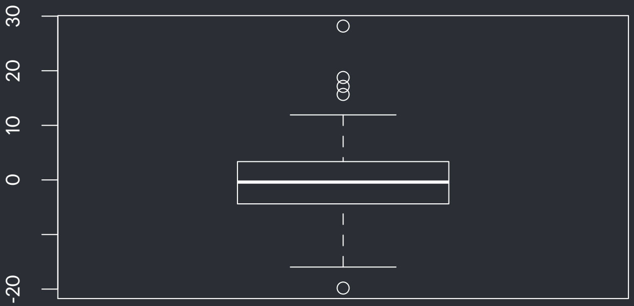
```
      14        27        52        85       154 
18.76901 -19.81152  17.14133  15.66079  28.18007 
```

Basically, the model is not able to explain these cases very well. Let's also see them on the

plot.

Outliers among the residuals

```
boxplot(resid(model2))
```



Outliers among the residuals

```
  mclean[154,] # 154th row and all columns
```

```
# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <dbl>        <int>    <chr>
1     95          4        1            1 Internal
```

Someone with only 1 year of experience and poor rating is hired as manager at $95K!

Outliers among the residuals

```
mclean[154,] # 154th row and all columns

# A tibble: 1 5
  Salary MngrRating YearsExp YrsSinceGrad   Origin
   <int>      <dbl>    <dbl>        <int>    <chr>
1     95          4        1            1 Internal
```

Someone with only 1 year of experience and poor rating is hired as manager at $95K!

If you decide that this is an anomaly (e.g. CEO's son promoted!) that you don't want to include in your analysis, omit that row and report it in your conclusions.

Influential cases

The Residuals vs Leverage plot tells about influential cases.

Influential cases

The Residuals vs Leverage plot tells about influential cases.

An influential case is the one that could change your β values significantly when excluded from your analysis.

Influential cases

The Residuals vs Leverage plot tells about influential cases.

An influential case is the one that could change your β values significantly when excluded from your analysis.

In other words, they do not follow the overall trend.

Influential cases

The Residuals vs Leverage plot tells about influential cases.

An influential case is the one that could change your β values significantly when excluded from your analysis.

In other words, they do not follow the overall trend.

Look for the cases on the upper/lower right corners (beyond the dashed curves).