Learning Goals

1. Import a csv file
2. Clean up the names of the variables
3. Drawing a scatterplot with R
4. interpreting a scatterplot
5. Calculating the correlation coefficient in R
6. interpreting the correlation coefficient

Topic 2: Looking For Trends in Real Data, ScatterPlots and Correlations

hyperlinks are in blue

Working with data tends to get sanitized in statistics courses due to the fact that it is often difficult to download data from websites and it is rather time consuming and tedious to clean it up. For sports fans, current data is crucial and often it is available only on the web. For the sports fan who wants to move beyond the basic raw statistics and do some analysis getting and cleaning the data is crucial. In this section, we will download some data from the internet, and get a taste for what is involved in working with real data.

A Little Data Mining

The file `BBStatsA.csv` in Sakai is a csv file downloaded from the website `https://www.sports-reference.com/cbb/seasons/2018-school-stats.html` which is a good source for sports data. This file contains a list of all college basketball teams for the 2018-2019 season and various team statistics including a number of advanced basketball statistics. We have to do a little cleaning up on the file before importing it into R. We need to be mindful of a few differences in data presentation in R and in excel and we need to be aware of how data can get transformed in the import process. In this file, we will have to change the names of some columns to acceptable column names for R and we will need to check that the imported data is converted to data of the correct type. It is best to clean up as much as we can in excel first.

Names of Variables in R:

1. The data on the website looked like this

Advanced School Stats NCAA = Tournament Appearance Share & more ▾ Glossary

		Overall						Conf.		Home		Away		Points		School Advanced												
Rk	School	G	W	L	W-L%	SRS	SOS	W	L	W	L	W	L	Tm.	Opp.	Pace	ORtg	FTr	3PAr	TS%	TRB%	AST%	STL%	BLK%	eFG%	TOV%	ORB%	FT/FGA
1	Abilene Christian	17	14	3	.824	-3.52	-6.29	3	1	7	0	5	3	1291	1077	67.5	112.6	.347	.320	.584	51.4	57.1	13.2	8.0	.558	15.7	30.5	.243
2	Air Force	16	6	10	.375	-6.35	-0.42	1	3	5	3	0	5	1061	1125	66.5	98.1	.309	.387	.551	50.6	52.6	7.3	5.7	.529	20.2	24.9	.207
3	Akron	16	9	7	.563	6.17	-2.33	1	2	7	1	0	4	1173	1007	68.0	106.9	.322	.507	.537	49.8	51.0	7.7	8.5	.507	14.2	28.1	.224
4	Alabama A&M	17	2	15	.118	-19.22	-3.81	1	2	1	4	0	10	1022	1284	68.9	85.3	.259	.337	.459	46.6	52.9	11.2	4.6	.438	19.4	27.8	.154
5	Alabama-Birmingham	17	10	7	.588	-0.25	-1.85	2	2	8	1	1	3	1192	1130	67.5	103.1	.268	.345	.531	55.1	46.4	9.5	7.4	.512	16.9	34.4	.174
6	Alabama State	14	4	10	.286	-13.35	1.49	2	1	4	0	0	8	919	1021	70.7	92.8	.364	.391	.502	48.3	46.1	8.6	11.0	.478	19.3	27.5	.222

When we downloaded to Excel, the grouping labels in the top row are assigned to columns like so:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1			Overall						Conf.		Home		Away		Points	
2	Rk	School	G	W	L	W-L%	SRS	SOS	W	L	W	L	W	L	Tm.	Opp.
3	1	Abilene Chris	17	14	3	0.824	-3.52	-6.29	3	1	7	0	5	3	1291	1077
4	2	Air Force	16	6	10	0.375	-6.35	-0.42	1	3	5	3	0	5	1061	1125
5	3	Akron	16	9	7	0.563	6.17	-2.33	1	2	7	1	0	4	1173	1007
6	4	Alabama A&	17	2	15	0.118	-19.22	-3.81	1	2	1	4	0	10	1022	1284
7	5	Alabama-Bir	17	10	7	0.588	-0.25	-1.85	2	2	8	1	1	3	1192	1130
8	6	Alabama Sta	14	4	10	0.286	-13.35	1.49	2	1	4	0	0	8	919	1021
9	7	Alabama	15	10	5	0.667	9.05	5.45	1	2	6	2	1	2	1154	1100
10	8	Albany (NY)	17	5	12	0.294	-10.32	-5.14	0	2	3	5	2	5	1149	1215
11	9	Alcorn State	15	4	11	0.267	-21.33	-1.7	0	3	4	2	0	8	987	1048
12	10	American	15	8	7	0.533	-2.98	-7.52	2	2	5	2	3	5	1064	996
13	11	Appalachian	17	5	12	0.294	-4.75	1.61	0	4	5	2	0	6	1379	1339
14	12	Arizona State	16	11	5	0.688	10.52	4.08	2	2	7	2	2	2	1276	1173
15	13	Arizona	17	13	4	0.765	14.07	5.36	4	0	9	1	3	1	1272	1124
16	14	Little Rock	17	7	10	0.412	-2.99	-2.52	2	2	6	3	1	7	1331	1309
17	15	Arkansas-Pin	15	5	10	0.333	-13.84	-0.49	2	0	3	0	1	10	997	1151
18	16	Arkansas Sta	17	8	9	0.471	-6.84	-0.57	2	2	6	1	1	6	1279	1313
19	17	Arkansas	15	10	5	0.667	13.1	4.5	1	2	7	4	2	0	1194	1065
20	18	Army	17	7	10	0.412	-8.59	-3.96	2	2	5	2	2	6	1228	1251

	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1		School Advanced												
2		Pace	ORtg	FTr	3PAr	TS%	TRB%	AST%	STL%	BLK%	eFG%	TOV%	ORB%	FT/FGA
3		67.5	112.6	0.347	0.32	0.584	51.4	57.1	13.2	8	0.558	15.7	30.5	0.243
4		66.5	98.1	0.309	0.387	0.551	50.6	52.6	7.3	5.7	0.529	20.2	24.9	0.207
5		68	106.9	0.322	0.507	0.537	49.8	51	7.7	8.5	0.507	14.2	28.1	0.224
6		68.9	85.3	0.259	0.337	0.459	46.6	52.9	11.2	4.6	0.438	19.4	27.8	0.154
7		67.5	103.1	0.268	0.345	0.531	55.1	46.4	9.5	7.4	0.512	16.9	34.4	0.174
8		70.7	92.8	0.364	0.391	0.502	48.3	46.1	8.6	11	0.478	19.3	27.5	0.222
9		71.7	107.2	0.442	0.367	0.551	54.4	50	5.9	12.5	0.514	16.4	33.6	0.305
10		68.8	97.5	0.344	0.437	0.524	49.8	54.2	8.4	5.7	0.481	18.8	27.5	0.258
11		70.4	93.5	0.298	0.412	0.503	48.7	56.6	12.6	8.8	0.476	18.9	29.1	0.197
12		67.5	103.4	0.367	0.378	0.554	49.4	50.8	10.3	10.9	0.526	16.5	25.2	0.249
13		74.4	108.2	0.339	0.398	0.559	48.4	45.4	6.8	7.6	0.527	15.3	27.6	0.244
14		73.7	106.5	0.428	0.331	0.533	55.1	50.6	8.4	10.5	0.499	15.3	34.7	0.284
15		69.9	106.2	0.342	0.358	0.556	50.6	47.7	7.2	7.6	0.517	14.7	26.6	0.258
16		74.4	103.7	0.449	0.355	0.584	49.9	56.9	7.8	8.9	0.566	19.8	24	0.285
17		69.7	93.1	0.371	0.34	0.507	48.8	47.5	9.1	6.8	0.467	19	27	0.258
18		70.9	103	0.389	0.357	0.526	51.8	44.8	8.9	11.3	0.481	16.6	33.4	0.286
19		74.2	105.5	0.425	0.374	0.538	49.4	63.3	10.7	16.4	0.511	15.3	29.4	0.272
20		73.9	97.8	0.215	0.443	0.508	48.6	60.8	8.3	4.6	0.488	14.6	22.9	0.143

We want to amalgamate this double row of variable/column names into one before importing the data into R.

2. R has some restrictions on the names that you can attach to columns in your data-frame. We will change the variable names in the excel file to acceptable ones before importing to R. The following rules apply for writing Identifiers/names in R: (https://www.datamentor.io/r-programming/variable-constant/).

   (a) Identifiers can be a combination of letters, digits, period (.) and underscore (_).

   (b) It must start with a letter or a period. If it starts with a period, it cannot be followed by a digit.

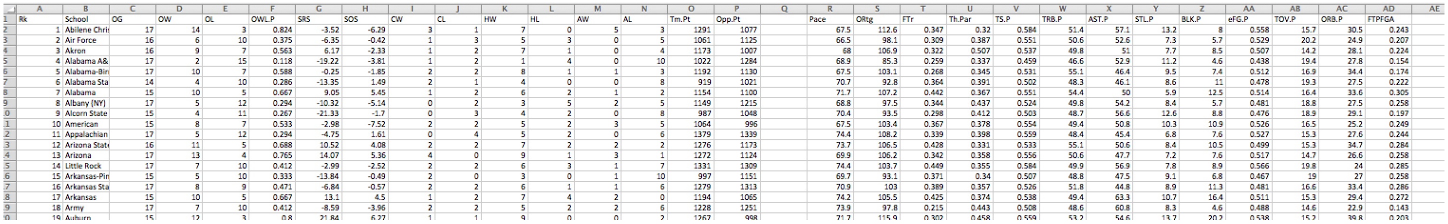
   (c) Reserved words in R cannot be used as identifiers.

   Clearly the name of any variable with a % in it must be changed, we'll change this to `.P`

New Labels Below, we show the original glossary and the new labels we have assigned to each column.

Rk – Rank
School – * = NCAA Tournament appearance
Overall
G – Games -> OG
W – Wins ->OW
L – Losses ->OL
W-L% – Win-Loss percentage -> OWL.P
SRS – Simple Rating System
A rating that takes into account average point differential and strength of schedule. The rating is denominated in points above/below average, where zero is average. Non-Division I games are excluded from the ratings.
SOS – Strength of Schedule
A rating of strength of schedule. The rating is denominated in points above/below average, where zero is average. Non-Division I games are excluded from the ratings.
Conf.
W – Conference Wins -> CW
L – Conference Losses ->CL
Home
W – Wins ->HW
L – Losses -> HL
Away
W – Wins ->AW
L – Losses -> AL
Points
Tm. – Points -> Tm.Pt
Opp. – Opponent Points -> Opp.Pt
School Advanced
Pace – Pace Factor
An estimate of school possessions per 40 minutes.
ORtg – Offensive Rating
An estimate of points scored (for teams) or points produced (for players) per 100 possessions.
FTr – Free Throw Attempt Rate
Number of FT Attempts Per FG Attempt
3PAr – 3-Point Attempt Rate -> Th.Par
Percentage of FG Attempts from 3-Point Range
TS% – True Shooting Percentage -> TS.P
A measure of shooting efficiency that takes into account 2-point field goals, 3-point field goals, and free throws.
TRB% – Total Rebound Percentage -> TRB.P
An estimate of the percentage of available rebounds a player grabbed while he was on the floor.
AST% – Assist Percentage -> AST.P
An estimate of the percentage of teammate field goals a player assisted while he was on the floor.
STL% – Steal Percentage -> STL.P
An estimate of the percentage of opponent possessions that end with a steal by the player while he was on the floor.
BLK% – Block Percentage -> BLK.P
An estimate of the percentage of opponent two-point field goal attempts blocked by the player while he was on the floor.
eFG% – Effective Field Goal Percentage -> eFG.P
; this statistic adjusts for the fact that a 3-point field goal is worth one more point than a 2-point field goal.
TOV% – Turnover Percentage ->TOV.P
; an estimate of turnovers per 100 plays.
ORB% – Offensive Rebound Percentage - > ORB.P
; an estimate of the percentage of available offensive rebounds a player grabbed while he was on the floor.
FT/FGA – Free Throws Per Field Goal Attempt -> FTPFGA

The new file now looks like this (It is called BBstats.csv in Sakai):



Rk	School	OG	OW	OL	OWL.P	SRS	SOS	CW	CL	HW	HL	AW	AL	Tm.Pt	Opp.Pt
1	Abilene Chri	17	14	3	0.824	-3.52	-6.29	3	1	7	0	5	3	1291	1077
2	Air Force	16	6	10	0.375	-6.35	-0.42	1	3	5	3	0	5	1061	1125
3	Akron	16	9	7	0.563	6.17	-2.33	1	2	7	1	0	4	1173	1007
4	Alabama A&	17	2	15	0.118	-19.22	-3.81	1	2	1	4	0	10	1022	1284
5	Alabama-Bir	17	10	7	0.588	-0.25	-1.85	2	2	8	1	1	3	1192	1130
6	Alabama Sta	14	4	10	0.286	-13.35	1.49	2	1	4	0	0	8	919	1021
7	Alabama	15	10	5	0.667	9.05	5.45	1	2	6	2	1	2	1154	1100
8	Albany (NY)	17	5	12	0.294	-10.32	-5.14	0	2	3	5	2	5	1149	1215
9	Alcorn State	15	4	11	0.267	-21.33	-1.7	0	3	4	2	0	8	987	1048
10	American	15	8	7	0.533	-2.98	-7.52	2	2	5	2	3	5	1064	996
11	Appalachian	17	5	12	0.294	-4.75	1.61	0	4	5	2	0	6	1379	1339
12	Arizona State	16	11	5	0.688	10.52	4.08	2	2	7	2	2	2	1276	1173
13	Arizona	17	13	4	0.765	14.07	5.36	4	0	9	1	3	1	1272	1124
14	Little Rock	17	7	10	0.412	-2.99	-2.52	2	2	6	3	1	7	1331	1309
15	Arkansas-Pin	15	5	10	0.333	-13.84	-0.49	2	0	3	0	1	10	997	1151
16	Arkansas Sta	17	8	9	0.471	-6.84	-0.57	2	2	6	1	1	6	1279	1313
17	Arkansas	15	10	5	0.667	13.1	4.5	1	2	7	4	2	0	1194	1065
18	Army	17	7	10	0.412	-8.59	-3.96	2	2	5	2	2	6	1228	1251
19	Auburn	15	12	3	0.8	21.84	6.27	1	1	9	0	0	2	1267	998

Rk	Pace	ORtg	FTr	Th.Par	TS.P	TRB.P	AST.P	STL.P	BLK.P	eFG.P	TOV.P	ORB.P	FTPFGA
1	67.5	112.6	0.347	0.32	0.584	51.4	57.1	13.2	8	0.558	15.7	30.5	0.243
2	66.5	98.1	0.309	0.387	0.551	50.6	52.6	7.3	5.7	0.529	20.2	24.9	0.207
3	68	106.9	0.322	0.507	0.537	49.8	51	7.7	8.5	0.507	14.2	28.1	0.224
4	68.9	85.3	0.259	0.337	0.459	46.6	52.9	11.2	4.6	0.438	19.4	27.8	0.154
5	67.5	103.1	0.268	0.345	0.531	55.1	46.4	9.5	7.4	0.512	16.9	34.4	0.174
6	70.7	92.8	0.364	0.391	0.502	48.3	46.1	8.6	11	0.478	19.3	27.5	0.222
7	71.7	107.2	0.442	0.367	0.551	54.4	50	5.9	12.5	0.514	16.4	33.6	0.305
8	68.8	97.5	0.344	0.437	0.524	49.8	54.2	8.4	5.7	0.481	18.8	27.5	0.258
9	70.4	93.5	0.298	0.412	0.503	48.7	56.6	12.6	8.8	0.476	18.9	29.1	0.197
10	67.5	103.4	0.367	0.378	0.554	49.4	50.8	10.3	10.9	0.526	16.5	25.2	0.249
11	74.4	108.2	0.339	0.398	0.559	48.4	45.4	6.8	7.6	0.527	15.3	27.6	0.244
12	73.7	106.5	0.428	0.331	0.533	55.1	50.6	8.4	10.5	0.499	15.3	34.7	0.284
13	69.9	106.2	0.342	0.358	0.556	50.6	47.7	7.2	7.6	0.517	14.7	26.6	0.258
14	74.4	103.7	0.449	0.355	0.584	49.9	56.9	7.8	8.9	0.566	19.8	24	0.285
15	69.7	93.1	0.371	0.34	0.507	48.8	47.5	9.1	6.8	0.467	19	27	0.258
16	70.9	103	0.389	0.357	0.526	51.8	44.8	8.9	11.3	0.481	16.6	33.4	0.286
17	74.2	105.5	0.425	0.374	0.538	49.4	63.3	10.7	16.4	0.511	15.3	29.4	0.272
18	73.9	97.8	0.215	0.443	0.508	48.6	60.8	8.3	4.6	0.488	14.6	22.9	0.143
19	71.7	115.9	0.302	0.458	0.559	53.2	54.6	13.7	20.2	0.538	15.2	39.8	0.203

We are now ready to import the file into *R*.

1. We store the file in the working directory/folder.

2. We use the command `read.csv()` to import the file. We can give the resulting import a name (say BBdata) in R, so that it will be stored as an object that we can retrieve in R. If we look under `help` at the command `read.csv`, we see that there are a number of options to be specified. The command:

   ```
   BBdata <- read.csv(file="BBStats.csv", header=TRUE, sep=",")
   ```

   reads the file `BBStats.csv` into a data frame that it creates called `BBdata`.
   `header=TRUE` specifies that this data includes a header row and
   `sep=","` specifies that the data is separated by commas (though read.csv implies the same it's safer to be explicit).

Viewing the data in R Top check that everything has transferred properly, you can click on the data frame `BBdata` under `Environment` in the upper right hand pane, or you can print out the first few rows with the command `head(BBdata, )`.

Data Exploration There are several ways to explore and use the data. One can compute statistics describing central tendency and variation in the variables. One can rank teams according to the values of various variables or combinations thereof. One can make plots and graphs. As a preliminary excursion, we will start with making a picture, called a scatterplot, to help us identify relationships between variables visually.

Scatterplots Here is an example of fantasy football points for 22 NFL quarterbacks for two consecutive seasons.

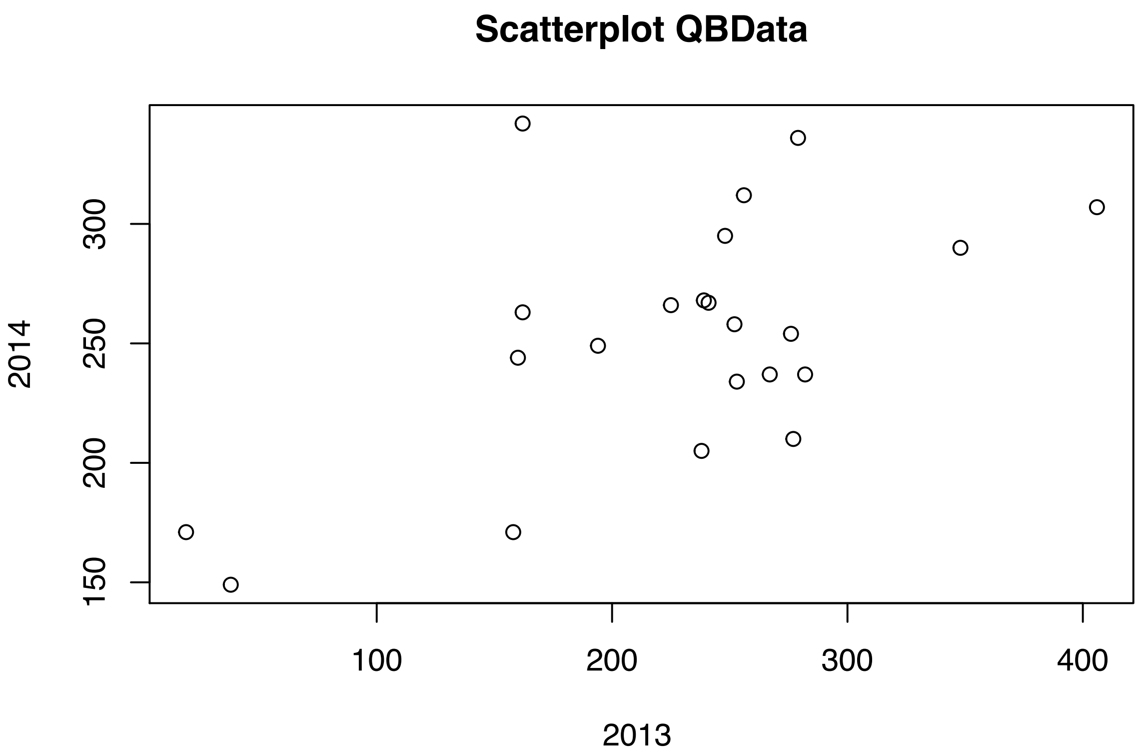
	2013	2014
Aaron Rodgers	162	342
Andrew Luck	279	336
Russell Wilson	256	312
Peyton Manning	406	307
Ben Roethlisberger	248	295
Drew Brees	348	290
Matt Ryan	239	268
Tom Brady	241	267
Ryan Tannehill	225	266
Eli Manning	162	263
Tony Romo	252	258

	2013	2014
Philip Rivers	276	254
Joe Flacco	194	249
Jay Cutler	160	244
Matthew Stafford	267	237
Cam Newton	282	237
Colin Kaepernick	253	234
Andy Dalton	277	210
Alex Smith	238	205
Kyle Orton	19	171
Ryan Fitzpatrick	158	171
Brian Hoyer	38	149

We would expect that if the "points earned" in a season is a "good statistic" in that it was a reasonable reflection of the quarterback's performance in a season, quarterbacks with relatively (relative to the other quarterbacks) high points in one season would earn relatively high points in the next season and vice-versa. We can make a picture called a scatterplot from the data, which helps us visualize the strength of the relationship between the points from season to season. We plot the points from the 2013 season on the horizontal axis and those from the 2014 season on the vertical axis. Each dot represents the pair of numbers (points for 2013, points for 2014) for one of the quarterbacks on the list. The resulting **scatterplot** for our data lis as follows:

Quarterback data:

```
Y2013<-c(162,279,256,406,248,348,239,241,225,162,252,
         276,194,160,267,282,253,277,238,19,158,38)
Y2014<-c(342,336,312,307,295,290,268,267,266,263,258,
         254,249,244,237,237,234,210,205,171,171,149)
plot(Y2013, Y2014,  main="Scatterplot QBData",
    xlab="2013 ", ylab="2014")
```



(note we specified the name of the plot and the labels on the axis along with the 2 data vectors when applying the `plot()` command.) We can see that indeed higher points in 2013 are "roughly" associated to higher points in 2014 from the picture. We may want to be more specific about the strength of this relationship and there are a number of statistical tools to help us measure the strength of the relationship some of which we will discuss below.

- If the dots (roughly) lie along a line sloping upwards then there is a positive relationship between them and higher points in 2013 is related to a higher number of points in 2014 and vice versa. A steeper slope to the line indicates a stronger relationship between the two variables.
- If the points on the scatterplot lie along a line sloping downwards there is a negative relationship between the 2 variables and higher points in 2013 are related to lower points in 2014.
- If the points lie along a horizontal line then all of the players got roughly the same number of points in 2014 no matter what the outcome in 2013.

- There is no linear relationship between the two variables if the points form a disc like shape or a nonlinear shape.

Pearson's Correlation Coefficient One way to measure the strength of the the linear relationship between two variables is with (Pearson's) correlation coefficient. Suppose our data is the set of pairs $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \quad \ldots \quad , (x_n, y_n)\}$. Then the correlation coefficient is given by

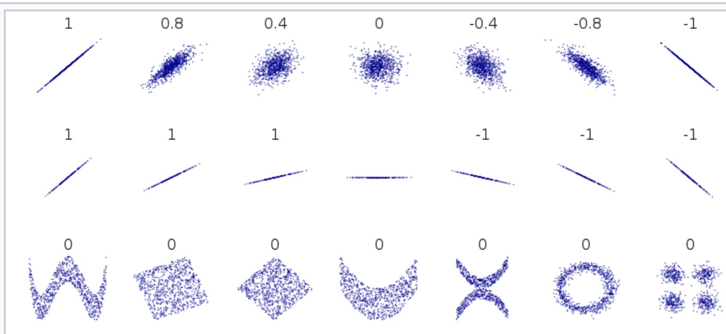
$$r = \frac{SS_{xy}}{\sqrt{SS_{xx}SS_{yy}}} \qquad \text{(as long as the denominator is not 0)}$$

where $SS_{xy} = \sum(x_i - \bar{x})(y_i - \bar{y}) = \sum x_i y_i - \frac{(\sum x_i)(\sum y_i)}{n}$,

$SS_{xx} = \sum(x_i - \bar{x})^2 = \sum x_i^2 - \frac{(\sum x_i)^2}{n}$, $SS_{yy} = \sum(y_i - \bar{y})^2 = \sum y_i^2 - \frac{(\sum y_i)^2}{n}$ and $\bar{x}$ and $\bar{y}$ are just the averages of the values of x and y respectively.

For those who have studied a little statistics: $r == \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y}$.

Interpretation of r The values of the correlation coefficient range between -1 and $+1$ Values close to 1 or -1 indicate a strong linear relationship between the variables with positive and negative slopes respectively. Values close to 0 indicate that there is almost no linear relationship between the variables. The following picture from wikipedia shows the value of the correlation coefficient for various scatterplots:



Several sets of (x, y) points, with the Pearson correlation coefficient of x and y for each set. Note that the correlation reflects the noisiness and direction of a linear relationship (top row), but not the slope of that relationship (middle), nor many aspects of nonlinear relationships (bottom). N.B.: the figure in the center has a slope of 0 but in that case the correlation coefficient is undefined because the variance of Y is zero.

If two variables are not related at all, the correlation coefficient will be 0, on the other hand, we may get a correlation coefficient of 0 for variables that are related in a non-linear fashion, such as the ones in the bottom row of the picture above. In general we have the following interpretation:

- $r = 1$ a perfect linear relationship with positive slope.
- $r = 0.7$ a strong (positive) linear relationship.
- $r = 0.5$ a moderate (positive) linear relationship.
- $r = 0$ no linear relationship

- $r = -0.5$ a moderate (negative) linear relationship.
- $r = -0.7$ a strong (negative) linear relationship.
- $r = -1$ a perfect linear relationship with negative slope.

Example Lets calculate the correlation coefficient for our variables X = scores for 2013 and Y = scores for 2014 in the quarterback data above. We use the inbuilt functions in R to calculate the averages (`mean()`) and the sums (`sum()`).

```
s1<-sum(Y2013)
s1
```

```
## [1] 4980
```

```
s2<-sum(Y2013)
s2
```

```
## [1] 4980
```

```
m1<-mean(Y2013)
m1
```

```
## [1] 226.3636
```

```
m2<-mean(Y2014)
m2
```

```
## [1] 252.9545
```

```
(Y2013 - m1) #Note this is a vector
```

```
##  [1]  -64.363636   52.636364   29.636364  179.636364   21.636364
##  [6]  121.636364   12.636364   14.636364   -1.363636  -64.363636
## [11]   25.636364   49.636364  -32.363636  -66.363636   40.636364
## [16]   55.636364   26.636364   50.636364   11.636364 -207.363636
## [21]  -68.363636 -188.363636
```

```
(Y2014-m2)
```

```
##  [1]   89.045455   83.045455   59.045455   54.045455   42.045455
##  [6]   37.045455   15.045455   14.045455   13.045455   10.045455
## [11]    5.045455    1.045455   -3.954545   -8.954545  -15.954545
## [16]  -15.954545  -18.954545  -42.954545  -47.954545  -81.954545
## [21]  -81.954545 -103.954545
```

```
(Y2013 - m1)*(Y2014-m2) #this is the product of corresponding entries in the above two vectors
```

```
##  [1] -5731.28926  4371.21074  1749.89256  9708.52893   909.71074
##  [6]  4506.07438   190.11983   205.57438   -17.78926  -646.56198
## [11]   129.34711    51.89256   127.98347   594.25620  -648.33471
## [16]  -887.65289  -504.88017 -2175.06198  -558.01653 16994.39256
## [21]  5602.71074 19581.25620
```

```
SSxy<-sum((Y2013 - m1)*(Y2014-m2)) #this is a number, the sum of the numbers in the above vector
SSxy
```

```
## [1] 53553.36
```

```
SSxx<-sum((Y2013 - m1)*(Y2013-m1))
SSxx
```

```
## [1] 159725.1
```

```
SSyy<-sum((Y2014 - m2)*(Y2014-m2))
SSyy
```

```
## [1] 54442.95
```

```
Pr<- SSxy/(sqrt(SSxx*SSyy))
Pr
```

```
## [1] 0.5742875
```

As with a lot of statistical functions, you can calculate the correlation of two vectors (named X and Y) with a single command in R `cor(X,Y)`:

```
cor(Y2013,Y2014)
```

```
## [1] 0.5742875
```

We see that we have a moderate to strong positive linear relationship between the variables.

Example Lets return to our imported data and create scatterplots for some team statistics and the Overall Win-Loss Percentage OWL.P

(a) Create a scatterplot of the pair of values of the variables Win-Loss Percentage `OWL.P` and Effective Field Goal Percentage `eFG.P` for each team in our file `BBdata` using the basic command `plot(x, y)`. Label the axes as `Effective Field Goal Percentage` and `Overall Win-Loss Percentage`

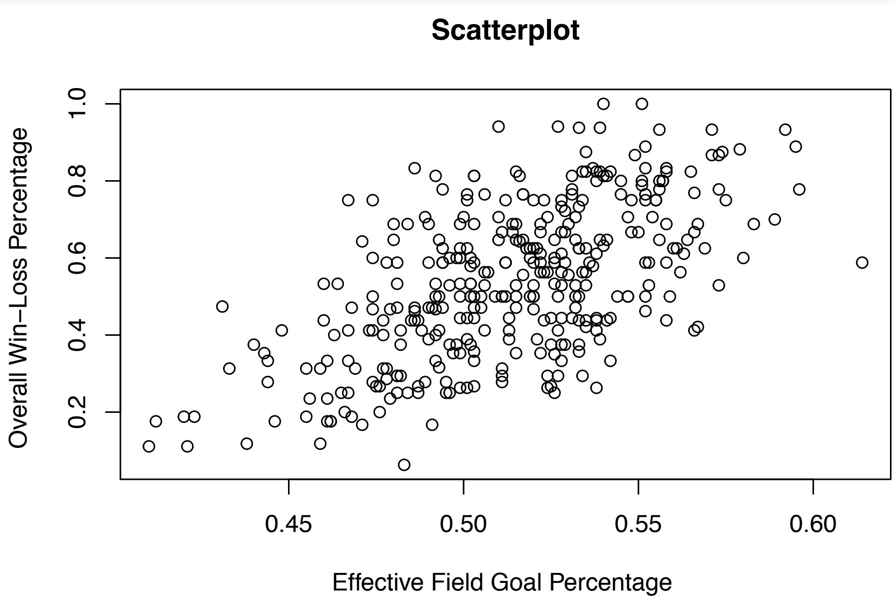
(b) Calculate the correlation coefficient for the variables in part (a)

(c) Create a scatterplot of the pair of values of the variables Win-Loss Percentage `OWL.P` and 3-Point Attempt Rate `Th.Par` for each team in our file `BBdata` using the basic command `plot(x, y)`. Label the axes as `Three Point Attempt Rate` and `Overall Win-Loss Percentage`

(d) Calculate the correlation coefficient for the variables in part (c)

(e) Which of the above two statistics would you prefer to use if you were trying to predict wins?

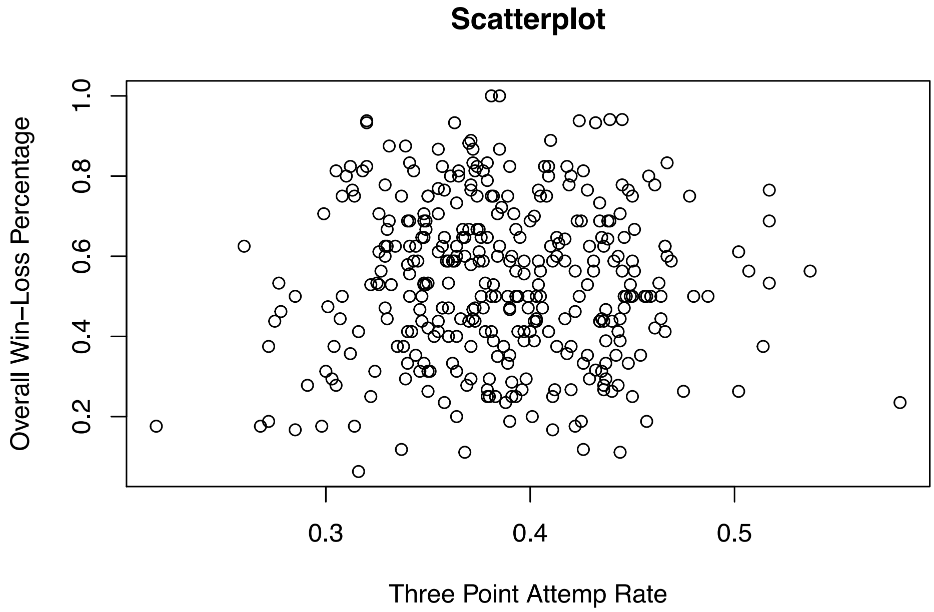
```
plot(BBdata$eFG.P, BBdata$OWL.P,  main="Scatterplot",
    xlab="Effective Field Goal Percentage ", ylab="Overall Win-Loss Percentage ")
```




```
cor(BBdata$eFG.P, BBdata$OWL.P)
```

```
## [1] 0.6002358
```

```
plot(BBdata$Th.Par, BBdata$OWL.P,  main="Scatterplot",
    xlab="Three Point Attemp Rate", ylab="Overall Win-Loss Percentage")
```




```
cor(BBdata$Th.Par, BBdata$OWL.P)
```

```
## [1] 0.004512129
```

R commands

1. `BBdata <- read.csv(file="BBStats.csv", header=TRUE, sep=",")` : import the csv file called `BBStats.csv`, use top line as variable names, call the imported version `BBdata` in R.
2. `head(BBdata,)` : shows first few rows of the data frame `BBdata`.
3. `head(BBdata, 20)` : shows first 20 rows of the data frame `BBdata`.
4. `plot(X,Y, main="MyPlot", xlab="x-values", ylab="y-values")` Creates a scatterplot for the data $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \quad \dots \quad , (x_n, y_n)\}$, where $X < -c(x_1, \dots, x_n)$ and $Y < -c(y_1, \dots, y_n)$. The name of the plot is `MyPlot`, the name on the horizontal axis is `x-values` and the name on the vertical axis is `y-values`.
5. `cor(X,Y)` gives the correlation between `X` and `Y`.

References

[1] Kubatko, J., Oliver, D., Pelton K., and Rosenbaum, D. *A Starting Point For Analyzing Basketball Statistics*. Journal of Quantitative Analysis in Sport, **Vol 3, Issue 3, 2007, Article 1.**

[2] Shea, Stephen M., and Baker, Christopher E. *Basketball Analytics.* Advanced Metrics, LLC, Lake St. Louis, MO, 2013.