1 Basic statistical models

Sample statistic	Distribution feature
Graphical	
Empirical distribution function F_n	Distribution function F
Kernel density estimate $f_{n,h}$ and histogram	Probability density f
(Number of X_i equal to a)/n	Probability mass function $p(a)$
Numerical	
Sample mean $\bar{X}_n$	Expectation μ
Sample median $\mathrm{Med}(X_1, X_2, \ldots, X_n)$	Median $q_{0.5} = F^{\mathrm{inv}}(0.5)$
pth empirical quantile $q_n(p)$	$100p$th percentile $q_p = F^{\mathrm{inv}}(p)$
Sample variance S_n^2	Variance σ^2
Sample standard deviation S_n	Standard deviation σ
$\mathrm{MAD}(X_1, X_2, \ldots, X_n)$	$F^{\mathrm{inv}}(0.75) - F^{\mathrm{inv}}(0.5)$, for symmetric F

Figure 1: Some sample statistics and corresponding distribution features

1.1 The linear regression model

To fit a linear regression model in R, we can use the `lm()` function, which uses the following syntax:

```
model <- lm(y ~ x1 + x2, data=df)
```

We can then use the following syntax to use the model to predict a single value:

```
predict(model, newdata = new)
```

The following examples show how to predict a single value using fitted regression models in R.

1.1.1 Simple linear regression model

The following code shows how to fit a simple linear regression model in R:

```
#create data
df <- data.frame(x = c(3, 4, 4, 5, 5, 6, 7, 8, 11, 12),
                 y = c(22, 24, 24, 25, 25, 27, 29, 31, 32, 36))

#fit simple linear regression model
model <- lm(y ~ x, data = df)
```

And we can use the following code to predict the response value for a new observation:

```
#define new observation
new <- data.frame(x = c(5))
# this has to be a dataframe otherwise it will not work.

#use the fitted model to predict the value for the new observation
predict(model, newdata = new)
```

```
##        1
## 25.36364
```

The model predicts that this new observation will have a response value of 25.36364.

1.1.2 Multiple linear regression model

Multiple Linear Regression Model

The following code shows how to fit a multiple linear regression model in R:

```
#create data
df <- data.frame(x1 = c(3, 4, 4, 5, 5, 6, 7, 8, 11, 12),
                 x2 = c(6, 6, 7, 7, 8, 9, 11, 13, 14, 14),
                 y = c(22, 24, 24, 25, 25, 27, 29, 31, 32, 36))

#fit multiple linear regression model
model <- lm(y ~ x1 + x2, data = df)
```

And we can use the following code to predict the response value for a new observation:

```
#define new observation
new <- data.frame(x1 = c(5),
                  x2 = c(10))
# here we can see why the predict function has to use a dataframe - because it
# can also be used to predict values of a multiple linear regression model

#use the fitted model to predict the value for the new observation
predict(model, newdata = new)
```

```
##        1
## 26.17073
```

The model predicts that this new observation will have a response value of 26.17073.

1.1.3 Plotting a regression model

You can use the R visualization library ggplot2 to plot a fitted linear regression model using the following basic syntax:

```
ggplot(data,aes(x, y)) +
  geom_point() +
  geom_smooth(method='lm')
```

The following example shows how to use this syntax in practice.

Suppose we fit a simple linear regression model to the following dataset:

```
#create dataset
data <- data.frame(y = c(6, 7, 7, 9, 12, 13, 13, 15, 16, 19, 22, 23, 23, 25, 26),
                   x = c(1, 2, 2, 3, 4, 4, 5, 6, 6, 8, 9, 9, 11, 12, 12))

#fit linear regression model to dataset and view model summary
model <- lm(y~x, data = data)
summary(model)
```

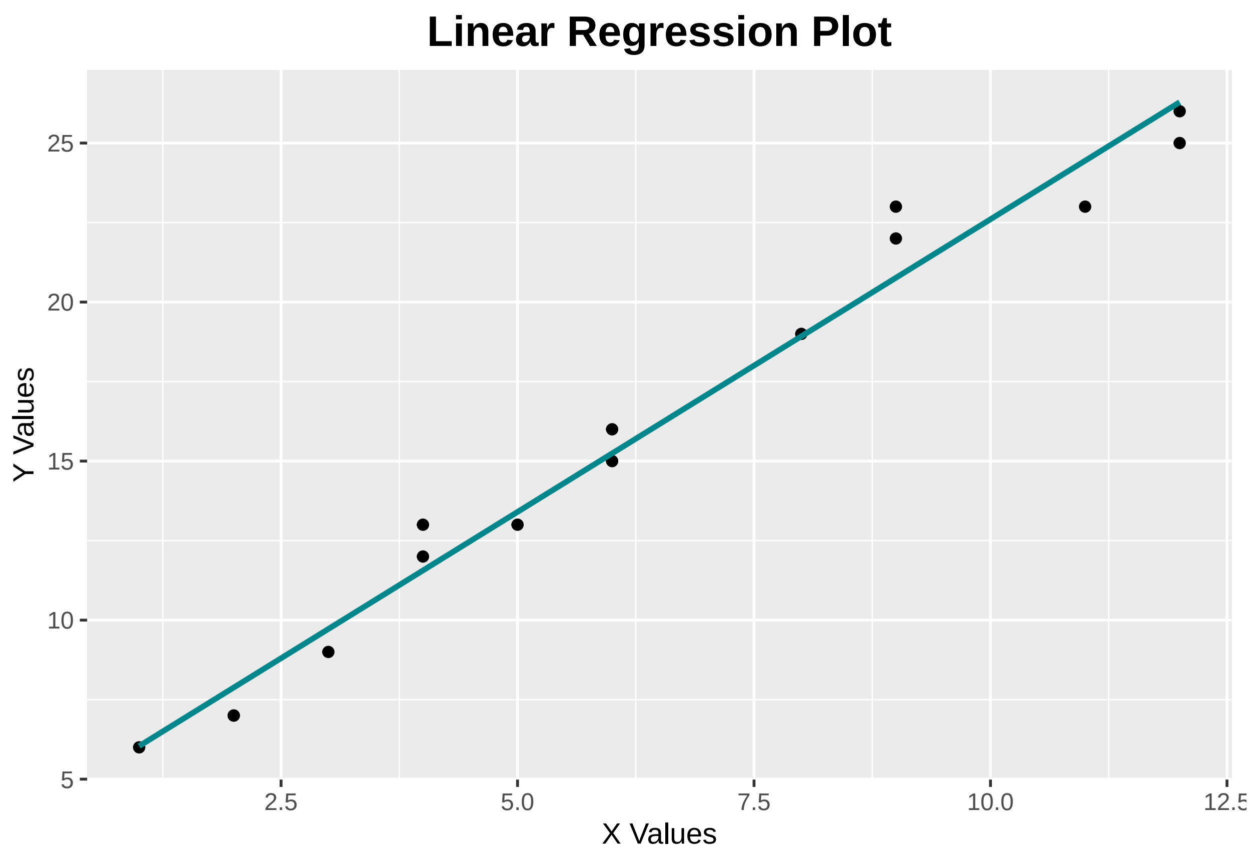
```
##
## Call:
## lm(formula = y ~ x, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4444 -0.8013 -0.2426  0.5978  2.2363
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.20041    0.56730   7.404 5.16e-06 ***
## x            1.84036    0.07857  23.423 5.13e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.091 on 13 degrees of freedom
## Multiple R-squared:  0.9769, Adjusted R-squared:  0.9751
## F-statistic: 548.7 on 1 and 13 DF,  p-value: 5.13e-12
```

```r
# make sure to import the ggplot library: library(ggplot2)

#create plot to visualize fitted linear regression model
ggplot(data, aes(x, y)) +
  geom_point() +
  geom_smooth(method = 'lm', se = FALSE, col = "turquoise4") +
  labs(x = 'X Values', y = 'Y Values', title = 'Linear Regression Plot') +
  #this last line makes the title header bold and in the middle
  theme(plot.title = element_text(hjust = 0.5, size = 16, face = 'bold'))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



Note the `se` argument removes the standard error from the visualization of if it is needed simply remove it or set it to `TRUE`