	Code	Interpretation
PACKAGES	library("dplyr") #need to call the libary before you use the package library("tidyr") library("rpivotTable") library("knitr")	
import excel	#import excel file into RStudio library(readxl) CR <- read_excel("Bank Credit Risk Data.xlsx", sheet = "Base Data", skip = 2)	
set	#check current working directory which is a 6x12 table getwd() #put in your working directory folder pathname #import excel file into RStudio library(readxl) Bank_Credit_Risk_Data <- read_excel("Bank Credit Risk Data.xlsx", sheet="Base Data", skip = 2) #path; skip 2 is to skip the first 2 rows of the excel BD<- Bank_Credit_Risk_Data head(BD) View(Bank_Credit_Risk_Data)	
data spreading	BD2.spread<- BD2 %>% spread(key=Gender,value=n) #long to wide BD2<- gather (data, key=Gender, value=n, -`Loan Purpose`) #wide to long	
Contingency Table	BD2 <- BD %>% group_by(`Loan Purpose`,Gender) %>% tally() #same as BD2 <- BD%>%count(`Loan Purpose`,Gender) BD2.spread<- BD2 %>% spread(key=Gender,value=n) BD2.spread[is.na(BD2.spread)]<-0 #convert NA to 0 value kable(BD2.spread, caption = "Contingency table for Loan Type & Gender") #or using rpivottable d1 <- gpa2[c("athlete", "female", "white", "black")] rpivotTable(d1, rows = c("athlete", "white", "black", 'female'), aggregatorName = "Count", height = 'auto') #SPECIFY HEIGHT to prevent overlapping of text.	#if no sub-category, can just put cols = c("Region") summarises the relationship between several categorical variables
Frequency Table	#dplyr Freq_type <- group_by(Data, Colkey) %>% summarise(Freq = n()) #base r Freq_type2 <- table(Data$Colkey) table_freq <- as.data.frame(Freq_type2)	
Relative Frequency Table	Freq.type$Rel.freq <- Freq_type$Freq/sum(Freq_type$Freq)	
Cumulative Relative Freq	cumfreq_table <- freq.table %>% mutate(cumfreq = cumsum(freq), cumrelfreq = cumfreq/nrow(data)	

	Code	Interpretation
Pie Chart	GenderFreq<-BD%>%count(Gender) kable(GenderFreq, caption = "Frequency of Bank Customers by Gender") slice.gen <- GenderFreq$n gen.piepercent <- 100*round(GenderFreq$n/sum(GenderFreq$n),2) label<-GenderFreq$Gender label<-paste(label,",",sep="") label<-paste(label,gen.piepercent) #default of sep=" " label<-paste(label,"%",sep="") pie(slice.gen,labels=label, col=c("blue","cyan"),radius=1, main="Customer Gender", density=20)	Should only be used when the number of categories are small
	#creating dataframe LoanFreq <- BD %>% count(`Loan Purpose`) kable(LoanFreq, caption = "Frequency Distribution for Loan Purpose") loanbar <- loanfreq$n barplot(loanbar, names.arg = Loanfreq$`Loan Purpose`, col = "blue", main = "Frequency of loan purpose", cex.names = 0.5, ylim = c(0,120), ylab = "no.of loans", las = 1, horiz = True)	plot thru data frame horiz = False (vertical barplot) las = 0 (vertical y-axis label) las = 1 (horizontal y-axis label)
Bar Chart	df <- fulldataset[,c(5,6,7)] value = as.matrix(df) colors <- c("green","orange" ,"brown","red","blue","yellow") Degrees <- c("Associate's","Bachelors","Advanced") barplot(value, names.arg = Degrees, col = colors, beside = TRUE , main = "Frequency of loan purpose", cex.names = 0.5, ylim = c(0,120), ylab = "no.of loans", las = 1, horiz = True)	beside = TRUE ## if it is a clustered bar plot: maritalstatus_vector <- c("Married", "Divorced", "Widowed") legend("topright", maritalstatus_vector, cex = 0.8, fill = colors_vector
Histogram	h1<-hist(BD$Age, main="Histogram of Customer Age",xlab="Customer Age", ylab="No. of Customers", col=c("darkorange"),xlim=c(10,80),ylim=c(0,100), labels=TRUE) #make table from histogram savings.group <- cut(BD$savings, h1$breaks, dig.lab=5), include.lowest = TRUE) t2 <- table(savings.group)	xaxp = c(0,20000,10) #changes the number of intervals for the x-axis label cex.axis = 0.8 probability = TRUE / labels = TRUE #take note that include.lowest is optional

	Code	Interpretation
Line Chart	v <- c(7, 12, 28, 3, 4) t <- c(14, 7,6,19,3) u <- c(3,5,20,31,40) plot(v, type = "o", col = "red", xlab = "month", ylab = "rain fall", main = "rain fall chart") lines(t, type = "o", col = "blue") lines(u, type = "o", col = "green") legend(1,40, legend = c("Region V", "Region T", "Region U"), col = c("red","blue"," green"), lty = 1, cex = 0.8	lty = 2 : dashed line magnitude of data values should not be far apart if not it will be hard to compare
Scatterplot	plot(BD$Age,BD$Savings, main="Scatterplot of Savings vs Age", xlab="Age", ylab="Savings",pch=6, ylim = c(6000, 13000), xlim = c(1250,2500))	show relationship between 2 variables
Pareto Analysis	#extract only the Savings column and sort in descending order BD.sav<-BD %>% select (Savings)%>% arrange(desc(Savings)) #compute the percentage of savings over total savings BD.sav$Percentage<-BD.sav$Savings/sum(BD.sav$Savings) #compute cumulative percentage for Savings BD.sav$Cumulative<-cumsum(BD.sav$Percentage) #compute cumulative percentage of customers from top most savings BD.sav$Cumulative.cust<-as.numeric(rownames(BD))/nrow(BD) which(BD.sav$Cumulative>0.8)[1] # compute percentage ofcustomers with top 80% savings which(BD.sav$Cumulative>0.8)[1]/nrow(BD)	To conduct pareto analyses on savings to understand what percentage of loan customers contribute to 80% of total savings amongst loan customers. Outcome: From the Pareto Analyses, we see that about 86 (out of 425) customers or 20% of the customers contribute to 80% of the total savings in the bank.
Computing quartiles	quantile(rmsize, c(0.25,0.5,0.75,1))	

	Code	Interpretation
PACKAGES	#install.packages("psych") #only need to run this code once to install the package library("dplyr") #need to call the libary before you use the package library("tidyr") library("rpivotTable") library("knitr") library("psych") library("RColorBrewer")	
Statistical Summary	# Manually generate using dplyr BD %>% summarise(vars=c("Age", "Savings"), n=n(), mean=c(mean(Age),mean(Savings)), # can add the descriptive statistic you need in the table on each line sd=c(sd(Age), sd(Savings)), median=c(median(Age),median(Savings)), min=c(min(Age),min(Savings)), max=c(max(Age),max(Savings)), IQR =c(IQR(Age),IQR(Savings)), skew=c(skew(Age),skew(Savings)), kurtosis=c(kurtosi(Age),kurtosi(Savings))) %>% mutate(across(where(is.numeric), round, 2)) %>% # to specify no. of decimal places kable(row.names=FALSE, caption = "Description Statistics for Age & Savings")	summary(x) describe(x) describeBy(data$col, group = data$grpuw)
	# Use describe() function in psych package to generate the descriptive statistics dfage <- describe(BD$Age, IQR=TRUE) dfsavings <- describe(BD$Savings, IQR=TRUE) df.desc1<-rbind(dfage,dfsavings) #df.desc1$trimmed <- df.desc1$mad <- df.desc1$se <- NULL # remove se, mad and trimmed if not needed df.desc1 <- df.desc1 %>% select(!c(trimmed, mad, se)) # remove se, mad and trimmed if not needed using dplyr select() df.desc1$vars<-c("Age","Savings") kable(df.desc1, row.names = FALSE, caption = "Descriptive Statistics for Age and Savings - using psych package")	
	# Create summary statistics of Age by loan purpose mat.ALP<-describeBy(BD$Age, group=BD$`Loan Purpose`, mat=TRUE, IQR=TRUE) mat.ALP<- mat.ALP %>% select(!c(item, vars, trimmed, mad, se)) # remove item, vars, trimmed, mad and se columns #mat.ALP<- mat.ALP[,-c(1,3,7,8,15)] # alternative way to remove item, vars, trimmed, mad and se columns kable(mat.ALP, caption = "Descriptive Statistics for Age grouped by Loan Purpose", row.names = FALSE)	

	Code	Interpretation
	# Manually generate each summary statistic (mean, sd, min, max, median) then combine into a table Age<-c(mean(BD$Age),sd(BD$Age),min(BD$Age),max(BD$Age),median(BD$Age)) Savings<-c(mean(BD$Savings),sd(BD$Savings),min(BD$Savings),max(BD$Savings), median(BD$Savings)) tab1<-rbind(Age,Savings) kable(tab1, row.names = TRUE, col.names = c("Mean","Std Dev", "Min","Max"," Median"), caption = "Descriptive Statistics for Age and Savings - manual")	
Plotting the cumulative frequency chart	plot(ecdf(x), main = "Cumulative Frequency of Computer Time", xlab = "Repair Time")	quantile(x, probs = 0.9) #gives u the value of repair time at 90%
Computing Mean	mean(x, trim = 0, na.rm = FALSE)	0 < trim < 0.5
Computing Median	median(x, na.rm = FALSE)	
Computing Mode	#use table func to obtain frequency value for each of X x <- data$`col` names(table(x))[table(x)==max(table(x))]	
Computing z-score	df$zscore <- (df$cost -mean(df$cost))/sd(df$cost)	measures how far an observation is from the mean
Coefficient of variation		CV = standard deviation / mean measures how volatile a data set is
Computing Skewness	cs.age<-(sum((BD$Age-mean(BD$Age))**3)/nrow(BD))/(sd(BD$Age))**3 ##can just use skew() from pysch	Symmetricalness of data Distributions that tail off to the right are called positively skewed; those that tail off to the left are said to be negatively skewed. relative symmetry 0.5 moderate 1 high degree of skewness Mean < Median < Mode (negative) Mode < Median < Mean (postive)
Computing Kurtosis	ck.age<-(sum((BD$Age-mean(BD$Age))**4)/nrow(BD))/((sd(BD$Age))**4) ##can just use kurtosis() from psych	If calculated using formula: CK < 3 indicates the data is somewhat flat with a wide degree of dispersion. CK > 3 indicates the data is somewhat peaked with less dispersion. else: consider 0
Computing Covariance	cov.AS<-cov(BD$Age, BD$Savings)	Measure of linear association between two variables X, Y Positive COV -> direct relationship Negative COV -> inverse relationship
Computing Correlation	cor.AS <- cor(BD$Age, BD$Savings)	#better than correlation because it is not affected by the unit of measurement Range: -1 (Strong negative) and 1 (Strong positive linear relationship) • interpretation of the magnitude 0 indicates no linear relationship; < 0.3 weak linear relationship; 0.3-0.7 moderate linear releationship; >0.7 strong linear relationship

	Code	Interpretation
Computing Proportion	length(data$supplier[data$supplier == "Spacetime Technologies"])/nrow(data) #however, for hypothesis testing, # compute z-statistic for proportion age50<- BD %>% filter(Age>50) p50 <- nrow(age50)/nrow(BD) z <- (p50 - 0.18) / sqrt(0.18*(1-0.18)/nrow(BD)) #compute critical value cv.age50<-qnorm(0.05) cv.age50	From our results (z-statistic=-3.98 & z-critical=-1.64), the z-statistic is lying in the lower critical region. Thus we have sufficient evidence to reject H0 and accept that proportion of Age is less than 0.18 at the 5% level of significance. computing the proportion of supplier being "Spacetime Technologies"
Box plot	# Plot boxplot to view the distribution of the data, and if outlier exists b1.5<- boxplot(D$Demand, horizontal=TRUE, xlab="Demand (1.5 IQR)", range=1.5) #applying 1.5*IQR to the left of Q1 or right of Q3 b3<-boxplot(D$Demand, horizontal=TRUE, xlab="Demand (3 IQR)", range = 3) #applying 3*IQR to the left of Q1 or right of Q3	
Outliers	# We can use the `out` component of the boxplot output to get the set of outliers in the data. b1.5$out # To create dataframes with and without outliers, we can use 1500 or b1.5$out[1] to filter D.outlier<- D %>% filter(Demand>=1500) # dataframe with just the outliers D.wo<- D %>% filter(Demand<1500) # dataframe without the outliers	
Shapiro Test	shapiro.test(D.wo$Demand)	W close to 1, p-value > 0.05 implies that the distribution of the data is not significantly different from normal distribution. In order words, the data does not deviate from normality
Proportion	d1<- D %>% filter(Demand.imp>800) pr1 <- nrow(d1)/nrow(D)	
Computing Probability	# we use the sample data here to estimate the mean and sd for demand. m<-mean(D$Demand.imp) s<-sd(D$Demand.imp) pr11<-pnorm(800,mean=m, sd=s, lower.tail = FALSE)	

	Code	**Interpretation**
PACKAGES		
	#compute manually 95% CI for mean Age uCIage95t<- mean(BD$Age) - qt(0.025,df=nrow(BD)-1)*sd(BD$Age)/sqrt(nrow(BD)) lCIage95t <- mean(BD$Age) + qt(0.025,df=nrow(BD)-1)*sd(BD$Age)/sqrt(nrow(BD)) print(cbind(lCIage95t, uCIage95t), digits=4)	
	#compute 95% CI for proportion (Age>50) n.bd=nrow(BD) age50<- BD %>% filter(Age>50) p50=nrow(age50)/nrow(BD) lCIp50 <- p50 + (qnorm(0.025)*sqrt(p50*(1-p50)/n.bd)) uCIp50 <- p50 - (qnorm(0.025)*sqrt(p50*(1-p50)/n.bd)) print(cbind(lCIp50, uCIp50),digits=3)	
Confidence interval	#compute 95% CI for mean Age using Rmisc::CI() ci.age<-CI(BD$Age, ci=0.95) ci.age print(cbind(ci.age[3],ci.age[1]), digits=4)	
Prediction interval	#chck for normal distribution plot(density(BD$Age),main="Density plot for Age") qqnorm(BD$Age, ylab="Sample Quantiles for Age") qqline(BD$Age, col="red") shapiro.test(BD$Age) #if normal mnage <- mean(BD$Age) sdage <- sd(BD$Age) n.bd <- nrow(BD) uPI.age <- mnage + (qt(0.995, df = (n.bd-1))*sdage*sqrt(1+1/n.bd)) lPI.age <- mnage - (qt(0.995, df = (n.bd-1))*sdage*sqrt(1+1/n.bd)) cbind(lPI.age, uPI.age) #else transform BD$lgage<-log10(BD$Age) #and then retest to chck if it is now normally distributed	

	Code	**Interpretation**
	#or transformTukey BD$Age.t = transformTukey(BD$Age, plotit=TRUE) #using -1 * x ^ lambda where lambda = -0.65 mnage.t <- mean(BD$Age.t) sdage.t <- sd(BD$Age.t) uPI.aget <- mnage.t + (qt(0.995, df = (n.bd-1))*sdage.t*sqrt(1+1/n.bd)) lPI.aget <- mnage.t - (qt(0.995, df = (n.bd-1))*sdage.t*sqrt(1+1/n.bd)) cbind(lPI.aget, uPI.aget) #reverse transform; comments below is to derive the formula # y=-1*x^lamda # -y = x^-0.65 = 1/(x^0.65) # x^0.65 = -1/y # x = (-1/y)^(1/0.65) lPI.age2 <- (-1/lPI.aget)^(1/0.65) uPI.age2<- (-1/uPI.aget)^(1/0.65) cbind(lPI.age2,uPI.age2) # reverse transform	
	ti<- (mean(BD$Age)-35)/(sd(BD$Age)/sqrt(nBD)) # t-value=-1.124 2*(pt(ti, nBD-1)) # p-value is 0.26153 or #using t.test function t.test(BD$Age, alternative="two.sided", mu=35, conf.level = 0.95)	mean Age of all their customers is 35 #alternative="less" or "greater"
One sample hypothesis test	t_stat <- qt(alpha , n-1) p-value <- pt(t_stat, n-1)	Comparing sample directly to population parameter at 95% level of significance -> alpha = 0.05
Two sample hypothesis test	##t.test(y~x) #y numeric; x factor t.test(usecar$Age ~ newcar$Age, alternative = "greater", data = dataset) ##t.test(y~x) #y numeric; x factor; variance == t.test(usecar$Age ~ newcar$Age, alternative = "greater", data = dataset, var.equal = TRUE) ##t.test(y,x) #y numeric; x numeric t.test(usecar$Age, newcar$Age) ##t.test(y,x) #y numeric; x numeric; paired t.test(usecar$Age, newcar$Age, paired = TRUE)	degrees of freedom = n1 + n2 - 2
F test (test for equality of variances)	var.test(y~x) critical f-value <- qf(.975, df1=7, df2=12) ##u get the df1 and df2 from var test output	must assume that both samples are drawn from normal samples if output F < F-critical, H0 cannot be rejected p>0.05 H0 cannot be rejected Hence, we conclude that there is no significant difference in variances at 5% level of significance.

	Code	Interpretation
ANOVA	# Since sample sizes are not equal, check if equal variance assumption is met fligner.test(Age ~ `Loan Purpose`,BD.less) #fligner.test gave p > 0.05. Hence we cannot reject H0 that variances are equal. Based on this result, we proceed to conduct ANOVA test. #normal anova test if all assumptions are met BD.less$`Loan Purpose`<-as.factor(BD.less$`Loan Purpose`) aov.age<-aov(BD.less$Age ~ BD.less$`Loan Purpose`) #note the group variable should be a factor summary(aov.age) TukeyHSD(aov.age) #If fligner test provides evidence that the variances are not equal, then we need to run the Welch ANOVA test, followed by Games Howell multiple comparison test. BD.less$Loan<-BD.less$`Loan Purpose` wa.out <- BD.less %>% welch_anova_test(Age ~ Loan) gh.out <- games_howell_test(BD.less, Age ~ Loan) # games howell test does not assume normality and equal variances wa.out gh.out	ANOVA Assumptions: 1) Randomly and independently obtained 2) Normally distributed (not that important) 3) Equal Variances (if the sample size the same, this is not important) *if sample size different and unequal variance, can use welch_anova_test(data,formula)

Functionality	Code	Interpretation
PACKAGES	library(dplyr) library(tidyr) library(ggplot2) # optional. we expect you to know base graphics, but allow ggplot if you find it easier library(wooldridge)	
Plots a scatterplot of y against x	plot(mroz$hours, mroz$wage, main=" Simple Scatterplot of wage vs. Hours", xlab="Working Hours", ylab="Wage")	
Estimating a Regression Model (single variable)	fit_wh = lm(wage ~ hours, data = mroz) summary(fit_wh) abline(fit_wh, col = 'red')	b0 - The mean value of Y when X is at level zero b1 - On average, for one-unit increase in X, Y increase / decreases by b1 unit. Degree of freedom: number of independent variable observations included in calculation R-square is 0.746 Model explains almost 75% of the total variation of Human Capital Index. p-value < 0.05: (if t value is very large and p-value is very small, the slope is significantly different from zero) sufficient evidence at 5% level of significance to reject the null hypothesis that: 1) pe is not statistically different from zero 2) all the slope predictors are zero and accept the alternative hypothesis that at least one of the coefficients of the slope predictors are not zero. thus, ... is a statistically significant predictor of Y
Estimating a Regression Model (multivariate)	fit_wh = lm(wage ~ hours + effort + skills, data = mroz) summary(fit_wh) abline(fit_wh, col = ’red’)	b0 - The mean value of Y when all X is at level zero b1 - On average, for one-unit increase in X1, Y increase / decreases by b1 unit, keeping all other variables constant.
Estimating a Regression Model (interaction)	fit_wh = lm(wage ~ hours + educ + hours*educ, data = mroz) summary(fit_wh) abline(fit_wh, col = ’red’)	Total Marginal Effect: Working one more hour increases average wage by b1 + b3 educ dollars, given the value of educ and holding all others constant. b1: Working one more hour directly increases average wage by b1 dollars, holding all other constant. b3: Working one more hour indirectly increases average wage by b3educ dollars, given the value of educ and all other constant.
Estimating a Regression Model (categorical independent variable)	fit_wh = lm(umbrella_sale ~ rainy + cloudy , data = mroz) summary(fit_wh) abline(fit_wh, col = ’red’)	Levels = {Sunny [0,0] , Rainy [1,0] , Cloudy [0,1]} b0 Average umbrella sale when it is sunny is b0 unit. b1 Average difference in umbrella sales when it is rainy compared to sunny is b1 unit. b2 Average difference in umbrella sales when it is cloudy compared to sunny is b1 unit. *as long as binary - interprete as dummy variable
Display the Regression Estimation Output	summary(fit_wh)	

Functionality	Code	Interpretation
Producing ANOVA table	anova_wh = aov(fit_wh) print(summary(anova_wh))	
Checking Assumptions of Linear Regression by plotting (residual plot and residual q-q plot)	resid_wh = resid(fit_wh) plot(mroz$hours, resid_wh, main="Residual Plot of resid vs. hours", xlab="Working Hours", ylab="Residuals") abline(0,0, lty = 'longdash') plot(fit_wh, 2)	First plot: Residual Plot (Biased, Heterokedasticity, Auto Correlation) Second plot: Residual Q-Q plot (Non-normal error)
Prediction of Linear Regression	new.mroz = data.frame(hours = c(2167, 975, 1790), educ = c(15, 11, 12), exper = c(12, 8, 3), expersq = c(12^2, 8^2, 3^2), city = c(1,0,1)) pred.fit_w = predict(fit_w, newdata = new.mroz, interval = " prediction") predci.fit_w = predict(fit_w, newdata = new.mroz, interval = " confidence") print(cbind(predci.fit_w, pred.fit_w))	Fit for confidence interval and Prediction interval is the same
Prediction of a point on OLS	new.mroz = data.frame(male = 1, kids = 1, age = 56, yrsmarr = 22, yrsmarrsq = 22^2, occup = 6, factorrelig = 'vry relig', factormarr = 'vry hap mar') pred.fit_w = predict(fit_3c, newdata = new.mroz) pred.fit_w	pred.fit_w = predict(fit_surv, newdata = new.mroz, type = 'response') *if logistic reg

Functionality	Code	Interpretation
PACKAGES	library(dplyr) library(tidyr) library(tseries) library(TTR) # One alternative for time-series in R library(forecast) # An alternative for time series in R library(car) # "Companion to Applied Regression" package, for F-test for linear combination of regression coefs library(wooldridge) # wooldridge data set will be used in this tutorial library(ggplot2) # optional. we expect you to know base graphics, but allow ggplot if you find it easier	
Running a logistics regression model	fit_surv = glm(survived ~ sex + age , family = binomial, data = titanic, control = list (maxit = 50)) summary(fit_surv)	b0: Log-odds when all X 's are zero. b1: Being a male decreases the log-odds of survival by |b1|, holding all other constant b2 Being each year older decreases the log-odds of survival by |b2|, holding all other constant.
Test for stationarity	adf.test()	
Creating a Time Series Object	fertil = ts(fertil3, frequency = 1, start = 1913)	gap of 1 year, start at 1913
Plotting a Time Series Line Graph	plot.ts(fertil[,'gfr'])	[,gfr] to choose the column
Plotting multiple Time Series Line Graph on the same axis	ts.plot(fertil[,'gfr'], fertil[,'pe_1'], gpars = list(xlab = 'Year', ylab = 'Value', col = c('darkred','darkblue'))) legend("topright", legend = c('gfr','pe_1'), col = c('darkred','darkblue'),lty = 1)	
Running regression for time series	fit_ip = lm(invpc ~ price, data = hseinv)	
Running regression for time series (with time variable / detrending)	fit_ipt = lm(invpc ~ price + t, data = hseinv)	Exclude spurious relattionships (where 2 variables conincidentally share the same pattern over time) Also, helps to detrend the time series to deal with the non-stationary issues
Plotting SMA	d1_long$ma4 = TTR::SMA(d1_long$PriceIndex, n = 4) d1_long$ma16 = TTR::SMA(d1_long$PriceIndex, n = 16) #base r plot plot(d1_long$TimeIndex, d1_long$PriceIndex, type="l", col="green", lwd=2, xlab= lines(d1_long$TimeIndex, d1_long$ma4, col="blue", lwd=2) lines(d1_long$TimeIndex, d1_long$ma16, col="red", lwd=2) #ggplot plot ggplot(d1_long, aes(x= TimeIndex)) + geom_line(aes(y=PriceIndex)) + geom_line(aes(y=ma4), col = "blue") + geom_line(aes(y=ma16), col = "red") + theme_bw()	n means the number of months
Getting the predicted value of SMA	sgfertil$ma4[length(sgfertil$ma4)]	The moving average the day before = predicted of the future

Functionality	Code	Interpretation
Exponential Smoothing Model (Triple Exponential Smoothing)	# split the souvenir into training set Jan87-Dec91 and test set Jan92-Dec93 souvenir_train = window(souvenirsale, start = 1987, end = c(1991,12)) souvenir_test = window(souvenirsale, start = c(1992,1), end = c(1993,12)) # train the HoltWinters on the training date souvenir_hw_train = HoltWinters(souvenir_train) # let's predict Jan1992-Dec1993 with the Holt-Winters model, return val for all n souvenir_pred_train = predict(souvenir_hw_train, n.ahead = 24) plot(souvenir_hw_train, souvenir_pred_train) lines(souvenir_test, col = "blue") # quantify the difference in terms of sum square errors sqrt(mean(souvenir_pred_train - souvenir_test)^2)	
Exponential Smoothing Model (Double Exponential Smoothing)	souvenir_hw_train = HoltWinters(souvenir_train, gamma = FALSE)	
Exponential Smoothing Model (Single Exponential Smoothing)	souvenir_hw_train = HoltWinters(souvenir_train, gamma = FALSE, beta = FALSE)	
Finding RMSE	rmse <- sqrt(mean(as.numeric((souvenir_pred_train[1:6]) - d1_wide_HELDOUT) rmse	[1:6], first to sixth data point
Plot Holt-Winters Pred and Actual on the same line	plot_min_value = min(c(souvenir_pred_train[1:6], unlist(as.vector(d1_wide_HELDOUT)))) plot_max_value = max(c(souvenir_pred_train[1:6], unlist(as.vector(d1_wide_HELDOUT)))) plot(1:6, souvenir_pred_train[1:6], type="l", col="red", ylim=c(plot_min_value, plot_max_value),xlab= "Time", ylab = "Price") lines(1:6, as.vector(d1_wide_HELDOUT), type ="l", col = "black") legend(x=1, y=135, legend=c("Holt-Winters", "Actual"), col=c("red","black"))	
ggplot (multiple line series on the same axis)	chick_set <- subset(ChickWeight, ChickWeight$Chick %in% c("3", "20", "24")) ggplot(chick_set, aes(x=Time, y=weight, group=Chick, colour=Chick)) + geom_line() + theme_bw()	#%in% boolean plots multiple time series on the same axis
ggplot (multiple line series on a few sets of axis)	ggplot(ChickWeight, aes(x = Time, y= weight, group = Chick, color = Chick)) + geom_line() + facet_grid(~Diet) + theme_bw() + theme(legend.position = "None")	facet_grid <- splitted grid for each diet group
Running linear regression with time series	set_diet1n3 <- subset(ChickWeight, ChickWeight$Diet %in% c("1","3")) %>% mutate(Dummy = factor(Diet, levels = c("1","3"),labels = c("1","3"))) fit_wh = lm(weight ~ Time*Dummy, data = set_diet1n3) summary(fit_wh)	#creating levels for dummy variable // reference level comes first in level to make it a factor #time*dummy bc dummy refers to the type of diet, and as time passes the effect of diet on weight becomes more significant. in time series, u need to consider how time plays a role in affecting ur variables
Describing Trend / Seasonality / Cyclicality		- Long Term Trend - Non - linear Trend - More volatile - Autocorrelation, thus non-stationary - cyclical: long-term pattern that shows fluctuations with no fixed interval e.g. inflation / recession - seasonal: pattern repeats at certain length of intervals

Functionality	Code	Interpretation		
PACKAGES	library(dplyr) library(tidyr) library(car) # for linearHypothesis() library(ggplot2) # optional. we expect you to know base graphics, but allow ggplot if you find it easier library(psych) # for pairs.panels() library(factoextra) # for fviz_cluster() library(wooldridge)			
Data Visualisation	data(iris) #load dataset pairs.panels(iris, lm=TRUE)			
F-test on model (drop one variable)	fit_restricted = lm(wage ~ educ + age + faminc + unem + city + exper + expersq, mroz) fit_unrestricted = lm(wage ~ hours + educ + age + faminc + unem + city + exper + expersq, mroz) anova(fit_restricted, fit_unrestricted)	A large F-statistic or a small p-value of such F-test shows strong evidence to reject the null hypothesis, (H0 : unrestricted model is not significantly better than restricted one in terms of explanatory power for Y .) Thus, slope for hours is statistically non-zero and unrestricted model is significantly better in terms of explanatory power.		
Stepwise model selection (auto choose which variables to drop) (backward)	step(model_full, direction = ’backward’)			
Stepwise model selection (auto choose which variables to drop) (forward)	model_intercept = lm(col_gpa ~ 1, data = gpa2) step(model_intercept, scope = ~ hours + age + educ + faminc + unem + city + exper + expersq, direction = ’forward’)	**use an empty model that only has b0 Scope refers to the list of potential predictors	Model selection	
Running PCA	pca_mroz = prcomp(formula = ~ . -wage -city, data = mroz, center = TRUE, scale = TRUE) summary(pca_mroz)	summarize information from all predictors except for wage (Y-variable) and city (categorical). ****must remove categorical variables**** output shows the proportion of variation explained along their dimensions		
Loadings of all 7 pcs	pca_mroz$rotation # examine the loading of PC1 and PC2 in first two columns of ’rotation’. rbind(pca_mroz$rotation[,1],pca_mroz$rotation[,2])	Loading refers to the coeffecient for each component variable in a PC.		
Check if result is correct	sum(pca_mroz$rotation[,1]^2)	result should equal 1		
Running Regression based on top k PCs	# extracting top 3 PC’s to run a linear regression of ’wage ~ pc1 + pc2 + pc3’ mroz_pca = mroz mroz_pca$pc1 = pca_mroz$x[,"PC1"] mroz_pca$pc2 = pca_mroz$x[,"PC2"] mroz_pca$pc3 = pca_mroz$x[,"PC3"] #linear pcafit = lm(wage ~ pc1 + pc2 + pc3, mroz_pca) #logistics classifier gpa <- glm(scholarship~ pc1 + pc2 + pc3, family = "binomial", data = gpa2)			
Show the composition of top 2 PCs w.r.t all other predictors	biplot(pca_mroz)	You can see how data points (grey) look like in the “plane” formed by pc1 and pc2 The red arrows in biplot are pointing in the direction of the original predictors	Data-dimensionality reduction	reducing features

Functionality	Code	Interpretation		
Determine the number of clusters (Within-cluster sum of squared distance)	set.seed(1) #starting point wss = rep(NA, 20) for (k in c(2:20)){ wss[k] = kmeans(whX, k, nstart = 10)$tot.withinss } plot(wss, type= "b", xlab = "Number of clusters", ylab = "Total within-cluster sum of squares")	Choose the elbow - value of k for 2 <= k <= 20		
k-means algorithm	km_mroz = kmeans(mroz, centers = 3, nstart = 10)		clustering	
applying PCA, then k-means algorithm	km_mroz = kmeans(mroz, centers = 3, nstart = 10) # k-means plot using 'fviz_cluster' in 'factoextra' package; fviz_cluster(km_mroz, data = mroz, palette = c("#00AFBB","#2E9FDF", "#FC4E07"), ggtheme = theme_minimal(), main = "Three clusters on the plane of first two PCs of 'mroz'.") km_mroz$centers #find the characteristics of each clusters	fviz_cluster() applies PCA first and plots the k-means clustering of observations that are projected onto the "plane" of top two PC's (x-axis PC1 - y-axis PC2)	clustering	
classification matrix	# use 'glm()' with specified parameter 'family = binomial' for logistic regression fit_surv = glm(survived ~ sex + age + sibsp + parch + fare + embarked, family = binomial, data = titanic, control = list(maxit = 50)) # predict the survival probability using fitted logistic regression predprob_surv = predict(fit_surv, type = 'response') # define survived = 1 when predicted probability >= 0.5; 0 otherwise pred_surv = ifelse(predprob_surv >= 0.5, 1, 0) # using 'confusionMatrix()' in 'caret' package cm = confusionMatrix(pred_surv, titanic$survived, positive = 'Survived')	#alternative base r tbl = table(gpa2$pred_scholarship, gpa2$scholarship)[2:1,2:1]	classification	For observations

	Actual: Yes	Actual: No
Predicted: Yes	true positive (TP)	false positive (FP)
Predicted: No	false negative (FN)	true negative (TN)

Error Types Table	H_0 is False	H_0 is True
Reject H_0	correct rejection $(1-\beta)$ true positive	type I error (α) false positive
Don't Reject H_0	type II error (β) false negative	correct no rejection $(1-\alpha)$ true negative

Sensitivity	$TP/(TP+FN)$	proportion of predicted yes out of actual yes
Specificity	$TN/(FP+TN)$	proportion of predicted no out of actual no
Precision	$TP/(TP+FP)$	proportion of actual yes out of predicted yes
Accuracy	$\frac{TP+TN}{TP+FN+FP+TN}$	proportion of correction classification
F1-Score	$\frac{2*Precision*Sensitivity}{Precision+Sensitivity}$	balanced index between precision and sensitivity

<table>
<tr><th>Functionality</th><th>Code</th><th>Interpretation</th></tr>
<tr><td>PACKAGES</td><td>library(lpSolve)</td><td></td></tr>
<tr><td>Table Formatting</td><td>Maximize total profit using decision variables X_1, X_2 | Profit = 0.15 X_1 + 0.40 X_2
--- | ---
Subject to |
Budget Constraint | 0.20X_1 + 0.70X_2 $\leq$ 100
Space Constraint | X_1 + X_2 $\leq$ 200
Non-Negativity Constraint 1 | X_1 + $\quad$ $\geq$ 0
Non-Negativity Constraint 2 | $\quad$ + X_2 $\geq$ 0</td><td></td></tr>
<tr><td>Computing the linear problem</td><td>objective_function = c(0.15, 0.40)
constraint_mat = matrix(c(0.20, 0.70, 1, 1), ncol = 2, byrow = TRUE)
constraint_dir = c('<=', '<=')
constraint_rhs = c(100, 200)
then solve the linear problem using 'lp()' function
lp_solution = lp(direction = "max", objective_function,
        constraint_mat, constraint_dir, constraint_rhs,
compute.sens = TRUE)
display the solution of linear problem: 'lp_obj$solution'
print(lp_solution$solution)
display the value of objective function at optimal solution
print(lp_solution)</td><td>Optimal solution is: X1=47.6, X2=0.00, X3=71.43. That is, run Factory A for 47.6 days, Factory B for 0 days and Factory C for 71.4 days. The Minimum cost is $154761.90. With the current constraints, the optimal solution involves not operating Factory B at all.</td></tr>
<tr><td>Identifying binding coefficients</td><td>num_cars <- sum(lp.solution$solution*c(30,40,50)</td><td>if close to / at the limit = binding</td></tr>
<tr><td>Sensitivity interval where the curr solution remains optimal</td><td>range_objcoef = cbind(lp_solution$sens.coef.from, lp_solution$sens.coef.to)
rownames(range_objcoef) = c('x1','x2');colnames(range_objcoef) = c('from','to')
print(range_objcoef)</td><td>It means that the curr solution is still optimal, as long as coef on x1 lies between [0.114, 0.400]
(interval refers to the coefficient of the variables in the objective function)</td></tr>
<tr><td>Compute shadow price</td><td># display shadow prices of constraints in sensitivity analysis
print(lp_solution$duals)</td><td>result follows order to the constraint_mat
shadow price: the marginal change in the optimal objective function value when the RHS of a constraint is increased by 1 #relax
if val of shadow price = 0 -> non-binding , else: binding
Shadow price is negative because if we allow Factory 3 (the most cost-efficient factory) to have more than 60 days, we can reduce our overall cost.
last few are non-negativity constraints, thus all zero.</td></tr>
</table>

Functionality	Code	Interpretation
PACKAGES	library(lpSolve)	
Computing integer optimisation (linear program relaxation)	# first define all parameters objective.fn = c(250, 225, 300) const.mat = matrix(c(7, 5, 8, 15, 30, 40, 1, 0, 0), ncol = 3, byrow = TRUE) const.dir = c("<=", "<=", "<=") const.rhs = c(60, 200, 7) # then solve model lp.solution = lp("max", objective.fn, const.mat, const.dir, const.rhs, int.vec = c(1,2,3), compute.sens = FALSE)	the only diff btw this and normal one is the int.vec (vector position of the decision variables)

Functionality	Code	Interpretation
Binary Decision Variables	const.mat = matrix(c("each city served by one center" rep(c(1,0,0,0,0), 4), rep(0,4), rep(c(0,1,0,0,0), 4), rep(0,4), rep(c(0,0,1,0,0), 4), rep(0,4), rep(c(0,0,0,1,0), 4), rep(0,4), rep(c(0,0,0,0,1), 4), rep(0,4), # constraint 6: "budget for at most 2 centers" rep(c(0,0,0,0,0), 4), rep(1,4), # constraints 7-26: "an open center for served cities" rep(0,0), -1, rep(0,19), rep(0, 0), 1, rep(0, 3), rep(0,1), -1, rep(0,18), rep(0, 0), 1, rep(0, 3), rep(0,2), -1, rep(0,17), rep(0, 0), 1, rep(0, 3), rep(0,3), -1, rep(0,16), rep(0, 0), 1, rep(0, 3), rep(0,4), -1, rep(0,15), rep(0, 0), 1, rep(0, 3), rep(0,5), -1, rep(0,14), rep(0, 1), 1, rep(0, 2), rep(0,6), -1, rep(0,13), rep(0, 1), 1, rep(0, 2), rep(0,7), -1, rep(0,12), rep(0, 1), 1, rep(0, 2), rep(0,8), -1, rep(0,11), rep(0, 1), 1, rep(0, 2), rep(0,9), -1, rep(0,10), rep(0, 1), 1, rep(0, 2), rep(0,10), -1, rep(0,9), rep(0, 2), 1, rep(0, 1), rep(0,11), -1, rep(0,8), rep(0, 2), 1, rep(0, 1), rep(0,12), -1, rep(0,7), rep(0, 2), 1, rep(0, 1), rep(0,13), -1, rep(0,6), rep(0, 2), 1, rep(0, 1), rep(0,14), -1, rep(0,5), rep(0, 2), 1, rep(0, 1), rep(0,15), -1, rep(0,4), rep(0, 3), 1, rep(0, 0), rep(0,16), -1, rep(0,3), rep(0, 3), 1, rep(0, 0), rep(0,17), -1, rep(0,2), rep(0, 3), 1, rep(0, 0), rep(0,18), -1, rep(0,1), rep(0, 3), 1, rep(0, 0), rep(0,19), -1, rep(0,0), rep(0, 3), 1, rep(0, 0)), ncol=24 , byrow=TRUE) objective.fn = c(40, 11, 75, 70, 60, 72, 77, 120, 30, 75, 24, 44, 45, 80, 90, 32, 55, 90, 20, 105, 0, 0, 0, 0) const.mat = matrix(c(...) , ncol=24 , byrow=TRUE) const.dir = c(rep(">=",5), "<=", rep(">=", 20)) const.rhs = c(rep(1,5), 2, rep(0,20)) # solve model lp.solution = lp("min", objective.fn, const.mat, const.dir, const.rhs, binary.vec = c(1:24)) # display optimal decision for where to build distribution centers lp.solution$solution[21:24]	

Functionality	Code	Interpretation

Logical Statement	Alternative	Integer Constraint
If A, then B	$B \geq A$	$B - A \geq 0$
If not A, then B	$B \geq (1 - A)$	$A + B \geq 1$
If A, then not B	$(1 - B) \geq A$	$A + B \leq 1$
At most one A or B	$A + B \leq 1$	$A + B \leq 1$
If A, then B and C	$B \geq A$ & $C \geq A$	$B + C - 2A \geq 0$
If A and B, then C	$C \geq (A + B - 1)$	$A + B - C \leq 1$

	Code	Interpretation
``{r q1c, echo = TRUE} ```		