## EXP 2:mean,median,mode

**File → Options → Add-ins → Go → Tick Analysis ToolPak → OK**
**Data → Data Analysis → Descriptive Statistics → OK**
**Select Input Range & Output Range**
**Tick Summary Statistics & Confidence Level → OK**

## EXP3: t-test,z-test,anova

**1. File → Options → Add-ins → Go → Tick Analysis ToolPak → OK**
**2. Data → Data Analysis → Z-Test: Two Sample Means → OK**
**3. Select Input Range for Variable 1 & Variable 2**
**4. Enter Mean Values (0.5 each)**
**5. Choose Output Range → OK**

**EXP4: missing data**
**1. Install / Enable → PrimaXL Add-in**
**2. PrimaXL → Choose "Missing"**
**3. Select Input Range (Marks with Missing Values)**
***4.* Choose Option → *Average Fill* or *Random Pick***
**5. Select Output Range → OK**

**EXP4: normalization**
**Use Formulas:**
**MAX() → Find Maximum**
**MIN() → Find Minimum**
**Find Difference: =`MAX - MIN`**
**Apply Normalization Formula:**
**=`(Value - MIN) / (MAX - MIN)`**
**Identify Best (Highest) Normalized Value**
**Format & Display Results**

**EXP5:pca,kpca**
**Load packages-Load dataset-Standardize data-Apply PCA-Plot PCA|**
**Apply Kernel PCA-Plot Kernel PCA-Apply SVD-Plot SVD-Compare results**

**EXP 5:**

```
install.packages("ggplot2")
install.packages("kernlab")
install.packages("ggfortify")
library(ggplot2)
library(kernlab)
library(ggfortify)
data(mtcars)
df <- scale(mtcars)
pca_result <- prcomp(df, scale. = TRUE)
summary(pca_result)
autoplot(pca_result, data = as.data.frame(df),
loadings = TRUE,
loadings.label = TRUE,
loadings.colour = "blue",
loadings.label.size = 4,
main = "PCA Biplot - mtcars (Enhanced)")
kpca_result <- kpca(~., data = as.data.frame(df), kernel = "rbfdot")
kpca_df <- as.data.frame(rotated(kpca_result))
kpca_df$Car <- rownames(df)
ggplot(kpca_df, aes(x = V1, y = V2, label = Car)) +
geom_point(color = "steelblue", size = 3) +
geom_text(vjust = -0.5, size = 3) +
theme_minimal() +
labs(title = "Kernel PCA (First 2 Components) - mtcars", x = "PC1", y = "PC2")
svd_result <- svd(df)
print(svd_result$d)
svd_df <- data.frame(U1 = svd_result$u[,1], U2 = svd_result$u[,2], Car = rownames(df))
ggplot(svd_df, aes(x = U1, y = U2, label = Car)) +
geom_point(color = "darkgreen", size = 3) +
geom_text(vjust = -0.5, size = 3) +
theme_minimal() +
labs(title = "SVD Components - mtcars", x = "U1", y = "U2")
```

**Exp6:bivariate,multivariate**
**Load iris dataset-Select numerical variables-Do correlation analysis-Plot scatterplots-Create scatterplot matrix-Perform MANOVA for species differences**

**EXP6:bivariate,multivariate**

```
data(iris)
df <- iris[, -5]
cor_matrix <- cor(df)
print(cor_matrix)
plot(df$Sepal.Length, df$Petal.Length, col=iris$Species,
main="Bivariate Analysis", xlab="Sepal Length", ylab="Petal Length")
pairs(df, col=iris$Species, main="Multivariate Scatterplot Matrix")
fit <- manova(cbind(Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) ~ Species, data=iris)
summary(fit)
```

**EXP7:plotting func**
**Load ggplot2 package-Load airquality dataset-Remove missing values → na.omit()-Plot Histogram → Temperature-Plot Boxplot → Ozone-Plot Density Curve → Wind-Plot Bar Chart → Month-Plot Scatter Plot → Ozone vs Temperature (color by Month)**

**EXP7:plotting func**

```
install.packages("ggplot2")
library(ggplot2)
data(airquality)
df <- na.omit(airquality)
hist(df$Temp, main="Histogram of Temperature",
col="skyblue", border="white", xlab="Temperature")
boxplot(df$Ozone, main="Boxplot of Ozone", col="orange")
plot(density(df$Wind), main="Density of Wind Speed", col="darkgreen", lwd=2)
month_count <- table(df$Month)
barplot(month_count, main="Barplot of Months",
col=c("red","blue","green","purple","cyan"))
ggplot(df, aes(x=Temp, y=Ozone, color=factor(Month))) +
geom_point(size=3) +
theme_minimal() +
labs(title="Scatter Plot (Ozone vs Temp by Month)", color="Month")
```

**EXP8:powerbi**
**Open Power BI-Get Data → Excel/CSV-Select sheet → Load-Clean data (headers, types, remove nulls)-Close & Apply Open Model View-Check tables-Create relationships-Verify data-Ready for visuals**

**EXP 9:DAX cal**
**Load student_data into Power BI**
**Check columns & data types, remove unwanted ones**
**Create DAX measures – Total Marks, Average Marks, Highest Score**
**Add visuals to design the report**
**Save and export the dashboard**

**EXP10:dashboard**
**Load financials dataset into Power BI**
**Check columns & data types, remove unwanted ones**
**In Report View, add required visuals**
**Create DAX measure – Profit_Per_Unit**
**Use Analyze to find key factors**
**Arrange visuals on one page**
**Publish report to workspace**
**View dashboard in Power BI Service**